
Memristor-assisted Background Calibration for Analog-to-Digital Converters (ADCs)

Author Zhaoguang Si
Submitted December 14th 2025

Supervisors Dr. Christos Papavassiliou¹
Dr. Shiwei Wang²
Prof. Timothy Constandinou³

¹Department of Electrical and Electronic Engineering, Imperial College London, UK.
(Supervisor)

²School of Engineering, University of Edinburgh, U.K.
(Co-supervisor)

³Department of Electrical and Electronic Engineering, Imperial College London, UK.
(Co-supervisor)

A thesis submitted in fulfilment of the requirements for the degree of *Philosophiae Doctor*

Circuits and Systems, Department of Electrical and Electronic Engineering,
Imperial College London

Circuits and Systems,
Department of Electrical and Electronic Engineering,
Imperial College London,
South Kensington,
SW7 2AZ, UK

Zhaoguang Si © 2025

Man takes his law from the Earth;
The Earth takes its law from Heaven;
Heaven takes its law from the Tao.
The law of the Tao is its being what it is.

Lao Tzu - *Tao Te Ching*

To my beloved family...
whose unwavering support has been a cornerstone since the
inception of my journey.
I am eternally grateful.

Declaration

I hereby declare that the work presented in this thesis is my own unless otherwise stated. To the best of my knowledge the work is original and ideas developed in collaboration with others have been appropriately referenced.

Zhaoguang Si

Copyright

The copyright of this thesis rests with the author. Unless otherwise indicated, its contents are licensed under a Creative Commons Attribution-Non Commercial 4.0 International Licence (CC BY-NC). Under this licence, you may copy and redistribute the material in any medium or format. You may also create and distribute modified versions of the work. This is on the condition that: you credit the author and do not use it, or any derivative works, for a commercial purpose. When reusing or sharing this work, ensure you make the licence terms clear to others by naming the licence and linking to the licence text. Where a work has been adapted, you should indicate that the work has been changed and describe those changes. Please seek permission from the copyright holder for uses of this work that are not included in this licence or permitted under UK Copyright Law.

In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

Zhaoguang Si

Acknowledgements

First and foremost, I would like to express my profound gratitude to Dr. Christos Papavassiliou and Dr. Shiwei Wang, whose mentorship has been instrumental throughout both my master's and doctoral studies. Their patient guidance and invaluable insights during my master's project not only led to significant achievements but also gave me the confidence to embark on my doctoral journey. I am particularly grateful for the PhD opportunity they provided during the challenging times of the pandemic, which marked the beginning of my career as an IC design researcher. Under their thoughtful mentorship and consistent encouragement, I have developed into a proficient IC design engineer, acquiring both technical expertise and professional acumen.

I am deeply indebted to my colleagues who have been integral to my research journey. Special thanks to Chaohan Wang, whose collaboration and mentorship in developing my technical skills has resulted in numerous successful projects. Our mutual trust and productive partnership have been cornerstone elements of my PhD journey. I extend my heartfelt appreciation to Xiongfei Jiang, my dear friend since our master's studies, whose unwavering support and encouragement have been invaluable. I am particularly grateful to Zheyi Li for imparting crucial professional skills that enabled me to complete one of the most significant components of my doctoral research. I would also like to acknowledge Mrs. Wieslawa Hsissen (Wiesia), our exceptional group administrator, whose efficient management and dedication have fostered an ideal research environment. Furthermore, I extend my appreciation to all team members at both Imperial College London and the University of Edinburgh for contributing to a highly motivated and inspiring research atmosphere.

My personal gratitude goes to Jue Wang, my girlfriend, whose understanding, patience, and selfless support have been a source of strength throughout this challenging journey.

Finally, and most profoundly, I wish to express my deepest gratitude to my family, especially my parents, whose unconditional love and unwavering support have been the foundation of my success. I am also thankful to my relatives for their constant care and for always keeping their doors open for me. Their presence and support have been fundamental to my academic achievements, and I am forever grateful for their role in my success.

Abstract

This thesis describes the main works undertaken to explore the feasibility of exploiting the memristor to improve the linearity of the ADC (i.e., integral non-linearity (INL), differential non-linearity (DNL), signal-to-noise-and-distortion ratio (SNDR), and spurious-free dynamic range (SFDR)). A memristor-assisted background calibration scheme has been proposed. Its functionality and efficacy have been demonstrated in Nyquist successive-approximation-register ADCs (SAR ADCs) with a resistive digital-to-analog converter (RDAC) and a capacitive DAC (CDAC). The complete circuitry is designed in a standard 180 nm and 65 nm Bipolar-CMOS-DMOS (BCD) process. The memristor model used for the circuit design was developed from a physical in-house fabricated memristor and validated in Cadence virtuoso simulation tools. At the end of the research, the proposed calibration scheme is able to calibrate a 100 kS/s 12b SAR ADC's SNDR from 49.40 dB to 63.70 dB, with a total ADC area of 0.08 mm^2 in 180 nm process, and a 5 MS/s 12b SAR ADC's SNDR from 46.23 dB to 66.10 dB, with a total ADC area of 0.0153 mm^2 in 65 nm process.

To the best of the author's knowledge, this is the first research on the memristor-assisted background calibration scheme for ADCs. As a feasibility study, this research is still at the initial stage, which means there are a lot of design challenges and limitations, making this emerging scheme is not yet competitive compared with the state-of-the-art ADCs with conventional CMOS calibration schemes. With the reported results, the author would like to draw attentions of the ADC design community to an alternative calibration scheme enabled by the emerging technology. The author expects the ADC performance will improve in the near future as the emerging CIM device technology (not only the "memristor") is advancing rapidly.

Contents

1	Introduction	19
1.1	Overview	20
1.1.1	Background and Motivation	20
1.1.2	Objectives	22
1.2	Thesis Outline	23
2	Calibration for Nyquist Data Converters & Memristor Devices	24
2.1	Memristor Devices	25
2.1.1	Background and History	25
2.1.2	Multi-State Memristive System State of the Art	26
2.1.3	CMOS/Memristor Hybrid System Design Challenges	28
2.1.3.1	Device-level Imperfections	28
2.1.3.2	Non-perfect Memristor Modeling	28
2.1.3.3	Circuit Design and Implementation Challenges	30
2.2	Calibration for Nyquist Data Converters	32
2.2.1	Historical Development of Calibration Techniques for ADCs	32
2.2.2	Bottlenecks in ADC Calibration for High-Resolution ADCs	33
2.2.3	Conclusion	35
3	Memristor-assisted Background Calibration for a 12-bit SAR ADC with RDAC	36
3.1	Introduction	37
3.2	The Concept of the Proposed Methodology	39
3.3	The Available Memristor and its Model	43
3.4	Implementation of the 12-bit SAR ADC with the Memristor-assisted Background Sign-based Calibration Scheme	45
3.4.1	SAR ADC Top-level Architecture	45
3.4.2	Implementation of the Bootstrapped Switch	47
3.4.3	Implementation of the Comparator	50
3.4.4	Implementation of the Asynchronous SAR Logic & Clock Generation	52
3.4.5	Implementation of the Detection & Code Alteration Circuit	54
3.4.6	Implementation of Sign Generator and Pulse Generator	56
3.4.7	Hybrid Binary/R-2R DAC	58
3.4.8	Code Alignment Circuit	60

3.5	Simulation Results of the proposed SAR ADC	62
3.5.1	Calibration Logic	62
3.5.2	Functionality of the DAC	63
3.5.3	Improvements in Static Linearity	64
3.5.4	Dynamic Performance and Walden Figure-of-Merit (FOM)	65
3.5.5	Power Consumption	67
3.6	Chip Layout	68
3.7	Testbench Design & Setup	70
3.7.1	Top-level System of the testbench	70
3.7.2	ADC interface design in Xilinx Zynq-7000	71
3.7.3	ADC evaluation board design	72
3.7.4	Testbench Integration	73
3.7.5	Measurement Challenges due to Memristor Device Limitations	74
3.8	Discussion and Conclusion	75
3.8.1	Discussion	75
3.8.2	Conclusion	76
4	Memristor-assisted Background Calibration for a 12-bit SAR ADCs with CDAC	77
4.1	Introduction	79
4.2	The Concept of the Proposed Methodology	83
4.3	Implementation of the 12-bit SAR ADC with the Memristor-assisted Background Sign-based Calibration Scheme - CDAC	87
4.3.1	SAR ADC Top-level Architecture	87
4.3.2	Implementation of the Comparator	88
4.3.3	Vcm-based Switching SAR Logic & Switching Control	89
4.3.4	Implementation of the Bootstrapped Switch	92
4.3.5	Implementation of full-custom MOM DAC arrays	95
4.3.6	Memrsitor-Assisted Voltage Generator	99
4.4	Floorplan of Proposed Design	100
4.5	Simulated FFT result of the Proposed SAR ADC	101
4.6	Discussion & Conclusions	102
5	Conclusions & Future Prospects	104
5.1	Conclusions	105
5.2	Publications & Contributions	106
5.3	Limitations & Challenges	107
5.4	Further Work & Future Prospects	108
	References	110

Appendices**116**

List of Figures

2.1	A common architecture of a 1T1R RRAM architecture.	30
2.2	One example of the conventional ADC capacitor mismatches calibration scheme.	34
3.1	Development of calibration techniques for data converters.	37
3.2	The concept of conventional calibration scheme.	39
3.3	Working principle of conventional correction circuits.	40
3.4	Memristor-assisted calibration scheme and its logic diagram.	41
3.5	The available memristor and its model.	44
3.6	Architecture of the 12-bit asynchronous SAR ADC.	45
3.7	Logic flow of the proposed SAR ADC shown in Figure 3.6	46
3.8	Circuit implementation of the bootstrapped switch.	47
3.9	FFT result of the bootstrapped switch.	48
3.10	ENOB histogram of the bootstrapped switch shown in Figure 3.8.	49
3.11	Dynamic comparator used in this design.	50
3.12	Histogram of the comparator input offset and its testbench.	51
3.13	Architecture of the SAR logic unit.	52
3.14	Clock generation control circuit and the overall asynchronous triggered clock generation circuitry.	53
3.15	Calibration activation codes for the DAC calibration and the implementation of the calibration activation code detection circuits.	54
3.16	Implementation of the code alteration circuitry.	55
3.17	The implementation of the sign generator and the programming pulse generator.	56
3.18	The 13-bit hybrid binary/R-2R DAC of the proposed 12-bit SAR ADC.	58
3.19	Function validation for code the alignment circuit.	60
3.20	Implementation of the code alignment circuit.	61
3.21	Simulation results to validate the calibration logic.	62
3.22	DAC function validation.	63
3.23	Improvements in Static linearity.	64
3.24	Dynamic performance of the proposed SAR ADC with memristor-assisted RDAC calibration.	65
3.25	Power consumption of each block in the ADC at 100 kS/s sampling rate.	67
3.26	(a) Layout of the core chip. (b) Layout of the chip for the tape out.	68
3.27	Microphotograph of the proposed design.	70
3.28	Microphotograph of the proposed design.	71
3.29	ADC interface implemented by Zynq 7000 processor.	71

3.30	PCB design of the ADC evaluation board.	72
3.31	Set up of the proposed data acquiring system.	73
4.1	Comparison between single-ended RDAC and differential CDAC.	80
4.2	Comparison among total switching energy consumed by different 12-bit SAR ADCs with different switching schemes.	82
4.3	Memristor-assisted calibration scheme for CDAC-based SAR ADC.	83
4.4	A 3-bit example of calibration logic of the proposed memristor-assisted CDAC background calibration for SAR ADCs.	85
4.5	Architecture of the proposed SAR ADC with a CDAC constructed by full-custom MOM capacitors.	87
4.6	Histogram of the comparator input offset (65 nm).	88
4.7	Vcm-based SAR Logic.	89
4.8	DAC switching control logic of the Vcm-based switching scheme.	89
4.9	The top-level circuitry of the internal comparator clock generation.	90
4.10	Timing diagram for the calibration function validation of the used Vcm-based switching scheme.	91
4.11	FFT result of the bootstrapped switch (65 nm).	92
4.12	Design method of MOM capacitor used for bootstrapped switch design.	93
4.13	Actual implementation of full-custom MOM capacitors used in the bootstrapped switch.	93
4.14	ENOB histogram of the bootstrapped switch (65 nm).	94
4.15	Histogram of smallest foundry-provided MOM capacitor matching.	95
4.16	Design method of the DAC built by unit-length MOM capacitors.	96
4.17	The full-customized MOM DAC layout design, using the method shown in Figure 4.16	97
4.18	The schematic of the V_{ref} generator.	99
4.19	The floorplan of the proposed design.	100
4.20	The FFT before and after calibration of the 12-bit SAR ADC with the mismatched CDAC:	101
5.1	Testbench for measuring the on-resistance of the bootstrapped switches.	116
5.2	Testbench for measuring the input offset of the comparator.	116
5.3	Linearity limitations of Delta Sigma modulator with multi-bit quantizers.	117

List of Tables

2.1	SUMMARY OF BEHAVIOUR OF MULTI STATE MEMRISTORS BASED ON DIFFERENT BI-LAYER STRUCTURES [38]	26
2.2	COMPARISON OF THE STATE-OF-THE-ART RRAM DESIGNS	27
3.1	Logic of the sign generator and pulse generator	57
3.2	Performance comparison	76
4.1	Look-up table for MSB calibration direction	86
4.2	MOM capacitor value extraction.	98
4.3	Look-up table for programming polarity	99
4.4	Performance comparison	102

List of Source Code

5.1	MATLAB code for ADC static linearity measurement.	118
5.2	VerilogA code for ramp generator.	119
5.3	MATLAB code for ADC dynamic linearity test [79].	120
5.4	ADC interface module Verilog code	123
5.5	Pin constraints for adc interface module	128
5.6	<i>adc_interface</i> module software control.	129

ADC Analog-to-Digital Converter
AI Artificial Intelligence
HDC Harmonic Distortion Correction
DNC DAC Noise Cancellation
BCD Bipolar-CMOS-DMOS
CDAC Capacitive Digital-to-Analog Converter
CIM Compute-in-Memory
CMOS Complementary Metal-Oxide-Semiconductor
DAC Digital-to-Analog Converter
DFF D Flip-Flop
DNC DAC Noise Cancellation
DNL Differential Non-Linearity
DSP Digital Signal Processing
EMI Electromagnetic Interference
ENOB Effective Number of Bits
FFT Fast Fourier Transform
F.O.M Figure of Merit
HDC Harmonic Distortion Correction
INL Integral Non-Linearity
LLM Large Language Model
LSB Least Significant Bit
MIM Metal-Insulator-Metal
MOM Metal-Oxide-Metal
MSB Most Significant Bit
NS Noise-Shaping
PCB Printed Circuit Board
PSRR Power-Supply-Rejection-Ratio
PVT Process-Voltage-Temperature
RDAC Resistive Digital-to-Analog Converter
RRAM Resistive Random-Access Memory
SAR Successive Approximation Register
SFDR Spurious-Free Dynamic Range

SNDR Signal-to-Noise-and-Distortion Ratio

SoC System-on-Chip

TSPC True Single-Phase Clock

UART Universal Asynchronous Receiver-Transmitter

VLSI Very-Large-Scale Integration

DWA Data Weighted Averaging

DEM Dynamic Element Matching

Chapter 1

Introduction

The explosion of data and the emergence of artificial intelligence enabled by machine learning (ML) is driving the development of “compute-in-memory” and other novel memory-intensive and memory-centric architectures that promise to overcome the “von Neumann bottleneck” – the energy inefficiency and high latency caused by shuttling data back and forth between separate memory and compute elements [1].

- *National Science and Technology Council, The White House, USA,*
March 2024

This introductory chapter presents the background, motivation, and objectives of this research. While it provides a comprehensive overview of the research topics addressed in each chapter, detailed discussions are presented within the individual chapters’ introductory sections. Ideas of this chapter were published in the ***IEEE Transactions on Circuits and Systems I: Regular Papers (Vol: 70, Issue: 9, Sep. 2023)*** [2] and ***IEEE Journal on Emerging and Selected Topics in Circuits and Systems (JETCAS), Volumn: 12, Issue: 4, December 2022.*** [3] © 2025 IEEE.

1.1 Overview

1.1.1 Background and Motivation

The continuous advancement of semiconductor technology has been predominantly driven by Moore's law over the past decades. However, as CMOS technology approaches its fundamental physical limits, the semiconductor industry faces increasing challenges in maintaining this trajectory of improvement. The pursuit of novel computing architectures, particularly those addressing the limitations of traditional von Neumann architectures, has become crucial for future technological advancement. This paradigm shift has led to growing interest in emerging devices, especially those capable of compute-in-memory (CIM) operations.

Among these emerging technologies, the memristor stands out as a particularly promising candidate. First theoretically proposed by Leon Chua in 1971 [4] as the fourth fundamental circuit element linking electric charge and magnetic flux, the memristor remained a theoretical concept for nearly four decades. The breakthrough came in 2008 when researchers at Hewlett-Packard (HP) Laboratory successfully fabricated the first physical memristor device [5], demonstrating resistance switching behavior correlating to the amplitude, frequency, and polarity of applied voltages. The physical realization of memristors opened new possibilities in integrated circuit design, offering non-volatile resistance states that can be controlled, feature sizes scaling down to 2nm [6], and compatibility with standard CMOS processes for monolithic integration. These characteristics, combined with low power consumption and the potential for both memory and computing functions in a single device, make memristors particularly attractive for next-generation integrated circuits.

Concurrently, in the domain of mixed-signal circuits, analog-to-digital converters (ADCs) have been reaching their performance limits despite advances in CMOS technology. While ADC performance has benefited from deeper technology nodes and architectural innovations, fundamental challenges persist, particularly in achieving high resolution. The primary limitation stems from intrinsic element matching issues, which become increasingly problematic as resolution requirements increase. For instance, achieving one additional bit of resolution in a capacitive DAC typically requires quadrupling the area to meet matching requirements.

Conventional techniques for ADC calibration have evolved significantly over time. Early solutions relied on laser trimming [7] and fuse trimming [8], which were expensive and irreversible. As CMOS technology advanced, digital calibration techniques emerged, offering more flexibility but introducing significant overhead. Background calibration methods were subsequently developed to enable continuous operation during calibration, and recent approaches have begun leveraging machine learning and advanced digital processing. However, these conventional calibration techniques introduce substantial power overhead from digital processing circuits, significant area occupation

from correction elements and memory, and increased conversion latency due to calibration procedures. The complexity of peripheral circuits for calibration control and inherent trade-offs between calibration accuracy and operational speed further complicate these approaches [9].

The convergence of these challenges in ADC design with the emergence of memristor technology presents an promising opportunity. The unique properties of memristors - particularly their ability to serve as both memory elements and programmable resistors - could potentially address fundamental limitations in ADC calibration. Integration of memristors into ADC architectures offers the potential for reduced calibration overhead through combined memory and trimming functions, more efficient area utilization through vertical (3D) integration, and novel calibration approaches leveraging the memristor's analog nature. Furthermore, the recent emphasis on CIM architectures for artificial intelligence applications has accelerated memristor technology development. As highlighted in the March 2024 National Strategy on Microelectronics Research by the White House, CIM devices are considered crucial for overcoming the von Neumann bottleneck and enabling next-generation computing systems [1]. This strategic focus suggests that memristor technology will continue to mature, potentially offering increasingly reliable solutions for mixed-signal applications (e.g., ADCs, sensors).

This thesis explores this convergence of memristor technology and ADC calibration, proposing novel approaches to leverage memristors' unique characteristics for improving ADC performance. While acknowledging the current limitations of memristor technology, this research aims to establish foundational techniques that can evolve alongside technological advances in the field. The investigation focuses on developing practical implementations that demonstrate the feasibility of memristor-assisted calibration while addressing the challenges and limitations inherent in this emerging technology.

1.1.2 Objectives

This research aims to:

- Explore the feasibility of exploiting memristors to improve ADC linearity (INL, DNL, SNDR, and SFDR).
- Develop and validate a memristor-assisted background calibration scheme for both RDAC and CDAC architectures.
- Demonstrate implementation in Nyquist successive-approximation-register ADCs (SAR ADCs) with RDAC and CDAC architectures.
- Validate designs through comprehensive simulations and physical implementation in standard 180 nm and 65 nm Bipolar-CMOS-DMOS (BCD) processes.

1.2 Thesis Outline

The thesis is structured as follows:

Chapter 2 presents a comprehensive review of memristor device development and ADC calibration techniques, examining their historical development, state-of-the-art implementations, and current challenges. Contents in this chapter further strengths the motivations for the proposed research.

Chapter 3 introduces a memristor-assisted background calibration scheme implemented in a 12-bit 100 kS/s asynchronous SAR ADC with RDAC in 180 nm technology. This first-of-its-kind implementation demonstrates the feasibility of using memristors to improve ADC linearity, with results showing significant improvements in SNDR and SFDR.

Chapter 4 extends the research to CDAC architectures, addressing limitations found in the RDAC implementation. The design achieves notable performance improvements while maintaining a compact layout in 65 nm technology.

Chapter 5 concludes by summarizing the research contributions, discussing current limitations, and outlining prospects for future development as memristor technology continues to advance.

This thesis represents the first research exploring memristor-assisted background calibration for ADCs. While still at an initial stage with various design challenges and limitations, this work aims to draw attention to alternative calibration schemes enabled by emerging technology, anticipating improved ADC performance as CIM device technology advances.

Chapter 2

Calibration for Nyquist Data Converters & Memristor Devices

This chapter presents a comprehensive review of two key technological domains relating to the proposed research topic: memristor device development and ADC calibration techniques. Building upon the research motivations established in the introduction chapter, the development and state-of-the-art implementations in both fields are examined independently. Existing literature reveals extensive research on ADC calibration challenges for Nyquist data converters, highlighting persistent limitations in conventional approaches. The emergence of memristor technology presents novel opportunities to address these fundamental challenges, potentially revolutionizing conventional ADC calibration methodologies. This technological convergence not only reinforces the research motivations previously discussed but also establishes a compelling direction for advancing Nyquist ADC calibration through memristor-assisted solutions. (Note: While this chapter provides a comprehensive overview of memristor technology and ADC calibration techniques at a foundational level, each subsequent core chapter contains its own focused literature review respected to specific research aspects. These chapter-specific reviews examine recent developments and technical challenges directly relevant to the particular memristor applications being investigated.)

Contents of this chapter was published in the **IEEE Journal on Emerging and Selected Topics in Circuits and Systems (JETCAS)**, Volumn: 12, Issue: 4, December 2022. [3] © 2025 IEEE. The author's work covers the memristor device measurement, data analysis, drawing figures and paper's writing.

2.1 Memristor Devices

2.1.1 Background and History

In the past decades, the semiconductor industry has steadily followed Moore's law by advancing to deeper technology nodes. In a common view of that CMOS technology nodes may eventually approach its physical limit, emerging devices beyond the CMOS has become increasingly crucial in semiconductor research [1]. The pursuit of novel computing architectures such as neuromorphic computing has been driven by the need for higher data scalability limited by the traditional von-Neumann architectures. The rapid technological advances of memristors promise to resolve these issues, improving the capability of electronics beyond what the CMOS cannot achieve alone.

Since Leon Chua first proposed the concept of the memristor in 1971 [4], many studies investigating memristor characteristics were reported [10–14], but the conceptual memristor was not linked to any physical devices in the following three decades. In the early 2000s, an increasing number of Metal-Oxide-Metal (MOM) devices with non-volatile resistive switching behaviour started to emerge, such as the ones in [15–18]. In 2008, HP lab first linked a non-volatile resistive switching device based on Pt- TiO_2 -Pt layered structure [5]. Since then, memristors have attracted increasing attention from the research community. The memristor's unique features such as non-volatility, small feature size (down to 2nm [6]), low power consumption, and compatibility with CMOS technology for monolithic integration render them good candidates for using in various applications [19–23].

Among the memristor's distinctive capabilities, its multi-state switching characteristic stands as particularly significant. Since the physical realization of memristors in 2008, their multi-state properties have attracted considerable research interest, as a single memristor can potentially replace multiple transistors while maintaining equivalent functionality [24]. However, practical implementation faces several fundamental challenges, including retention degradation, device variability across cycles and between devices, process-voltage-temperature (PVT) variations, and limitations in modeling accuracy. These technical hurdles, combined with the complexity of control circuitry and programming requirements, currently constrain the performance of memristor-based systems [25]. The emergence of Large Language Models (LLMs) and advanced artificial intelligence has renewed focus on Compute-In-Memory (CIM) architectures, where memristive devices show promise in overcoming traditional von Neumann bottlenecks and enhancing the capabilities of conventional CMOS electronics.

2.1.2 Multi-State Memristive System State of the Art

The existing memristor devices in different materials are shown in Table 2.1. The monolayer memristor exhibits the lowest number of states, smallest resistive switching range, and low HRS/LRS ratio. Memristors based on TiO_2 with Al_xO_y ¹, WO_x , and ZnO_x are able to exhibit promising multi-state capability. In particular, the $TiO_2+Al_2O_3$ bi-layer shows the highest number of states and has been leveraged in different integrated applications such as a memristive memory (RRAM) in [26].

Despite stand-alone memristors can support 10 to 47 states (Table 2.1), an essential issue is assessing this capability within a CMOS/Memristor system. The highest reported number of independent states in a memristive neural network is 32 [27], which is equivalent to five bits. When employing these states, multi-state memristors in an array are unreliable and experience high device-to-device and cycle-to-cycle variations, low yield rate, and limited precision [27], constraining their practical application.

The past decade have seen the emergence of fully integrated memristive systems, especially RRAM in neuromorphic computing [26, 28–37]. Among these works, the multi-state RRAM exhibits its advantage of storing more information in a single cell [26, 29, 34, 35]. Table 2.2 shows the state-of-the-art RRAM systems, and few observations are acquired. Although weight resolution in an RRAM system can reach up to 20-bit in [36], the highest reported number of states in a single RRAM cell is 5, which is equivalent to 2.3-bit information storage per cell [29]. In this work, error windows are defined between resistance ranges to avoid the collision. When writing to a desired state, write voltages are precisely controlled. In 2021, a fully integrated RRAM black exhibiting 3 independent states that can be employed to store 1.58 bit/cell of information was reported [34]. Among all the state-of-the-art memristive systems, most of them continue to rely on binary operation (RS and LRS) for reliable switching. However, driven by increasing demands for high-density data storage, research continues to focus on developing more stable multi-state devices.

Table 2.1: SUMMARY OF BEHAVIOUR OF MULTI STATE MEMRISTORS BASED ON DIFFERENT BI-LAYER STRUCTURES [38]

Bi-layer Oxide	TiO_2	TiO_2 + Al_xO_y	TiO_2 + Ta_xO_y	TiO_2 + WO_x	TiO_2 + HfO_x	TiO_2 + ZnO_x	TiO_2 + SiO_x
Number of States	10	47	37	33	23	15	13
LRS (k Ω)	12.5	36	16.8	55	17.6	42	15
HRS (k Ω)	15.3	85	20.2	162	25.6	91	16.6
Range (k Ω)	2.8	49	3.4	107	8	49	1.6
HRS/LRS Ratio	1.2	2.4	1.2	2.9	1.5	2.1	1.1
Equivalent Number of Bits	3.3	5.5	5.2	5	4.5	3.9	3.7

¹ Al_xO_y stands for Aluminum Oxide such as Al_2O_3

Table 2.2: COMPARISON OF THE STATE-OF-THE-ART RRAM DESIGNS

	ISSCC 18 [28]	VLSI 18 [26]	ISSCC 19 [29]	ISSCC 19 [30]	ISSCC 20 [31]	ISSCC 20 [32]	ISSCC 20 [33]	SSCL 20 [35]	CICC 21 [34]	ISSCC 21 [36]	ISSCC 21 [37]
Technology	65nm	14nm	130nm	55nm	130nm	130nm	22nm	90nm	40nm	40nm	22nm
Supply (Volt)	1.0 V	5.5 V	0.71-1.2 V	1.0 V	4.2 V	1.8 V	0.7-0.9 V	1.2 V	0.9 V	0.9 V	0.8 V
Multi-bit Weight Memristor Cell	No	Yes	Yes	No	No	No	No	Yes	Yes	No	No
RRAM Type	Digital	Digital	Digital	Digital	Analog	Analog	Digital	Digital	Digital	Digital	Digital
Array Size	512 x 256	n/a	256 x 16	256 x 512	784 x 784	256 x 256	512 x 512	128 x 64	256 x 256	256 x 256	1024 x 512
Array Type	1T1R	1S1R	1T1R	1T1R	2T2R	1T1R	1T1R	1T1R	1T1R	1T1R	1T1R
Max Input Resolution	1 bit	n/a	1 bit	2 bit	1 bit	1 bit	4 bit	1 bit	1 bit	8 bit	8 bit
Max Weight Resolution	Ternary	1.58 bit/cell	5.6 bit 2.3 bit/cell	3 bit	Ternary	Analog	4 bit	2 bit/cell	1.58 bit/cell	8 bit	8 bit
Max Output Resolution	3 bit	n/a	1 bit	3 bit	8 bit	1 bit	11 bit	1 bit	4 bit	20 bit	14 bit
Write Verification	No	No	Yes	No	No	No	No	Yes	Yes	Yes	No
Sensing mode	Current	Current	n/a	Current	Current	Voltage	Current	Voltage	Voltage	Voltage	Voltage

2.1.3 CMOS/Memristor Hybrid System Design Challenges

While multi-state memristors have demonstrated promising capabilities in various applications, the technology faces several fundamental challenges that affect its integration into practical systems. Device variability, process control limitations, programming accuracy, state retention degradation, and manufacturing yield remain significant concerns. This section provides a comprehensive analysis of these technological challenges and their implications for CMOS/Memristor hybrid system development, examining how device-level imperfections influence circuit design decisions and system architecture considerations (Some of the observations are from the measurement of our in-house fabricated memristor [39], and measurements were done by the ArcOne commercial measurement platform of the group [40]).

2.1.3.1 Device-level Imperfections

The memristor's multi-state switching specification in integrated circuits design is stricter. A properly defined state has to be accurately rewritten by the same pulse train which is often measured within $2-3\sigma$. This requires a "program and verify" approach to guarantee the switching behaviour, but it adds circuit complexity and causes more difficulties for designers. With the assistance of the ArcOne measurement platform [40], the observation is that memristors under test did not always reach the desired state, the successful forming and writing process are not always perfectly repeatable. This leads to some states becoming incompatible and cannot be considered as a stable state. Moreover, memristors are sensitive to the voltage above their threshold, and they can be damaged with high voltage when performing electroforming. Yet, there is not a constant threshold voltage observed that can be applied to all memristor devices. Thus, when designing the memristor, this device-to-device variation has to be considered to avoid damaging the memristors. Besides, "non-volatile" memristors are not non-volatile under all possible configurations. The actual retention depends on the forming process, pulsed amplitude, and the present resistance state (e.g., state with larger resistance is more volatile). This retention degradation behaviour has been discussed in [41], and needs further investigation. Therefore, in a CMOS/Memristor hybrid system design, retention degradation must be considered and properly solved with proper design methods. Although the retention has been investigated in [42] and methods to improve the stability such as the multi-level incremental step pulse with verify algorithm in [43], there are no commonly adopted methods to overcome the retention degradation behaviour. These device-level imperfections have increased the circuit design difficulties of CMOS/Memristor hybrid system.

2.1.3.2 Non-perfect Memristor Modeling

Accurately modeling memristor behavior represents a crucial prerequisite for circuit design and system integration. A large class of models in literature such as TEAM [44] and VTEAM [45] are deterministic models constructed as a coupled set of differential and algebraic equations. A typical formulation consists of:

$$\frac{ds}{dt} = f_u(u, s) f_w(s) \quad (2.1)$$

$$i(t) = G(s, u) \cdot u(t) \quad (2.2)$$

where s represents the internal modeling state variable, $f_u(u, s)$ describes the input sensitivity function, and $f_w(s)$ serves as a window function. The memristor conductance can be derived from the state variable governed by eq. 2.1 and the input $u(t)$ using an appropriate function $G(s, u)$. The current through the device $i(t)$ and voltage across it $u(t)$ are therefore related by eq. 2.2.

In such a model, a given input stimulus and an initial condition specify a unique deterministic trajectory in the state space, towards a terminal state. Upon reaching the terminal state, no further change in state occurs despite the application of input. In practice, these terminal states are defined by using suitable window functions. In most models the window functions simply bind the state variable into a valid range used for modeling. In more complex models such as [46], the window is a function of the state variable and the input stimulus. Therefore, this model can additionally describe input-dependent stable states to which the memristance saturates. Whilst this is useful in cases where the device can be allowed to reach a steady state, the models are not yet upgraded to include the transient volatility that describes the state of the device shortly after the stimulus is removed. In practice, all valid memristance values are not stable and the memristance typically decays to another more stable value. This behaviour can be modeled by extending the basic state evolution differential equation 2.1, dividing the state space into stable regions and unstable regions. Furthermore, relaxation dynamics can also be introduced to describe volatile behaviour from the unstable states towards the stable states. The modeling approach in [47] introduces a single stable state into the dynamics via the following addition to 2.1:

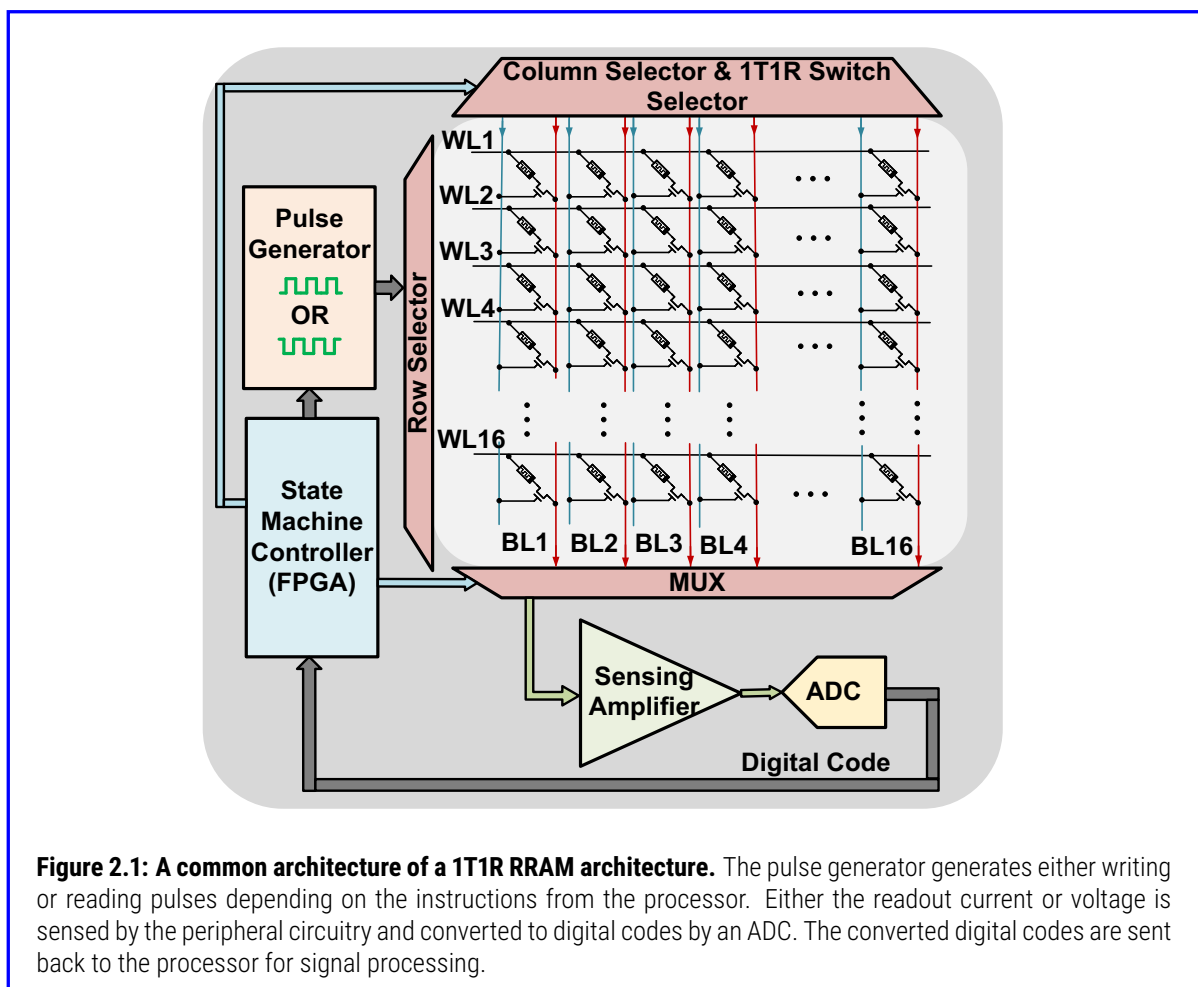
$$\frac{ds}{dt} = \left(f_u(u, s) - \frac{s - \epsilon}{\tau} \right) f_w(s) \quad (2.3)$$

In eq. 2.3, the only stable state is the value of s for which $\frac{ds}{dt} = 0$. Typically $f_u(u = 0, s) = 0$, thus the parameter ϵ is equal to the value of this single stable state in this model, while τ is the diffusion time constant. The models in [48, 49] extend this equation further by allowing the parameter ϵ to be a dynamic variable, governed by its own differential equation driven by the input stimulus. Therefore, in these models, the stable state of eq. 2.3 is variable and depends on the history of the input stimulus. Another approach to introduce a series of stable and unstable states in the governing dynamics is to impose a potential-energy-like function on the dynamics, whose minima correspond to stable states and maxima correspond to unstable states. For example, [50] introduces a combination of exponential functions and a sinusoid function to model the interfacial energy and periodicity of the transport dynamics respectively. Although the model accuracy issue caused by numerical integration of the model equations have been addressed [51, 52] and there

are ongoing efforts to develop accurate models for specific applications [53–55], the multi-state switching behaviour has not been modeled to the comprehensiveness required for robust and solid circuit design. To ensure a CMOS/Memristor hybrid system can work as expected, the memristor model needs to be extended more to cover switching metastability, temperature dependencies, and also the electroforming non-idealities.

2.1.3.3 Circuit Design and Implementation Challenges

Peripheral circuit design with multi-state memristor arrays is more challenging than in bi-state memristor arrays. Figure 2.1 illustrates the most common 1T1R architecture of the RRAM. When writing a memristor to the desired state, row and column selectors are used to select a device from the array. One terminal of the memristor is connected to a pulse generator and other terminals are connected to the ground or a virtual ground. The pulse generator then sends programming signals (e.g., pulses) to the selected memristor. The read operation faces more difficulties than the write operation. During the reading phase, low voltage pulses are applied to the selected memristor, while the resulting current is measured. The readout voltage has to be sufficiently low to avoid changing the memristance. The resulting current is interfaced with dedicated readout circuitry and gets digitized.



From Table 2.2, there are two types of sensing methods, the current mode and the voltage mode, both are adopted by the state-of-the-art designs. Current mode sensing provides higher speed and larger sensing margin [25]. However, current mode sensing is not preferred in a multi-state memristive system due to its higher noise nature compared to voltage mode sensing, leading to inaccurate readout result when the memristor has more states. Besides, the current mode sensing also means a large static power dissipation [25]. As for the voltage mode sensing, a current-to-voltage (I2V) circuitry is required. This I2V circuitry requires an amplifier with optimized dynamic range, input offset, input referred noise, open-loop gain and linearity because the multi-state memristor with high HRS/LRS ratio leads to a large current range. When the I2V conversion is finished, a high resolution ADC and a digital quantizer are needed to precisely categorize the I2V output voltage to different digital ranges corresponding to different distinguishable resistance state. More states that a 1T1R cell aims to achieve, the higher ADC resolution is needed. This leads to the trade-off among multi-state switching, power consumption, chip area, and circuit complexity. Additionally, to ensure the memristor is written to a desired state, a write-verify circuit is needed [29, 34–36], leading to a more sophisticated peripheral circuit design. Thus, there is a wide Pareto surface to optimize over and every bit of resolution requested from the memristive devices has to be carefully balanced against the costs of the corresponding peripherals.

2.2 Calibration for Nyquist Data Converters

Analog-to-digital converters (ADCs) are essential components in modern electronic systems, converting analog signals into digital data for processing. The accuracy and performance of ADCs are critical, especially in high-resolution applications such as wireless communication, medical imaging, and instrumentation. Over the years, calibration techniques have evolved significantly to enhance the performance of ADCs, particularly Nyquist ADCs. This section provides a comprehensive overview of the historical development, state-of-the-art techniques, and current challenges in ADC calibration, with a focus on recent advancements and bottlenecks.

2.2.1 Historical Development of Calibration Techniques for ADCs

The evolution of ADC calibration techniques represents a significant progression in mixed-signal circuit design methodology. Initial calibration approaches predominantly utilized foreground calibration techniques. During the 1970s, methodologies such as laser trimming [7] and fuse trimming [8] emerged as pioneering ADC calibration solutions, implementing calibration through physical modification of fundamental circuit elements in an off-chip manner. While these approaches achieved considerable linearities, they introduced substantial complexity in fabrication and packaging processes, limiting their practical implementation in modern integrated circuits.

A significant advancement occurred in 1998 [56] with the introduction of a continuously-calibrated 10 MS/s 12-bit ADC incorporating on-chip analog calibration. This innovation circumvented the necessity for complex signal processing. However, this calibration algorithm still necessitates interruption of the normal ADC operation, which is not desired by modern integrated circuit IPs. This limitation catalyzed research into background calibration techniques, enabling concurrent ADC operation during the calibration process. A notable implementation of this approach manifested in a time-interleaved ADC, utilizing monolithic analog background calibration to achieve gain and offset matching among component pipeline stages [57].

The proliferation of digital signal processing (DSP) capabilities, coupled with the continuous scaling of semiconductor technology nodes, precipitated a paradigm shift toward digital calibration methodologies. These techniques implemented digital algorithms for non-linearities correction, addressing key factors including gain error, comparator offset, and non-linearity. A significant advancement in this domain was the introduction of digital harmonic distortion correction (HDC) and DAC noise cancellation (DNC), specifically targeting residue amplifier gain error and DAC capacitor mismatches in pipeline ADCs [58]. While these digital approaches offered enhanced precision and automation, they imposed substantial overhead in terms of power consumption, silicon area, and conversion latency.

Contemporary applications, particularly in array-based interface applications such as radar systems, image sensors, and neural interfaces, have renewed interest in analog background calibration

techniques due to their superior efficiency in resource utilization [59]. Concurrently, noise-shaping (NS) techniques have emerged as effective solutions for mitigating mismatch effects within ADCs, thereby enhancing overall ADC performance metrics [60]. However, the implementation of NS systems introduces additional design complexity and requires supplementary peripheral circuits (e.g., residue amplifiers). Consequently, research in Nyquist ADC calibration techniques maintains its relevance in the field of mixed-signal design, driven by the continuous demand for efficient and practical solutions in modern integrated circuits, especially the large array-based applications.

2.2.2 Bottlenecks in ADC Calibration for High-Resolution ADCs

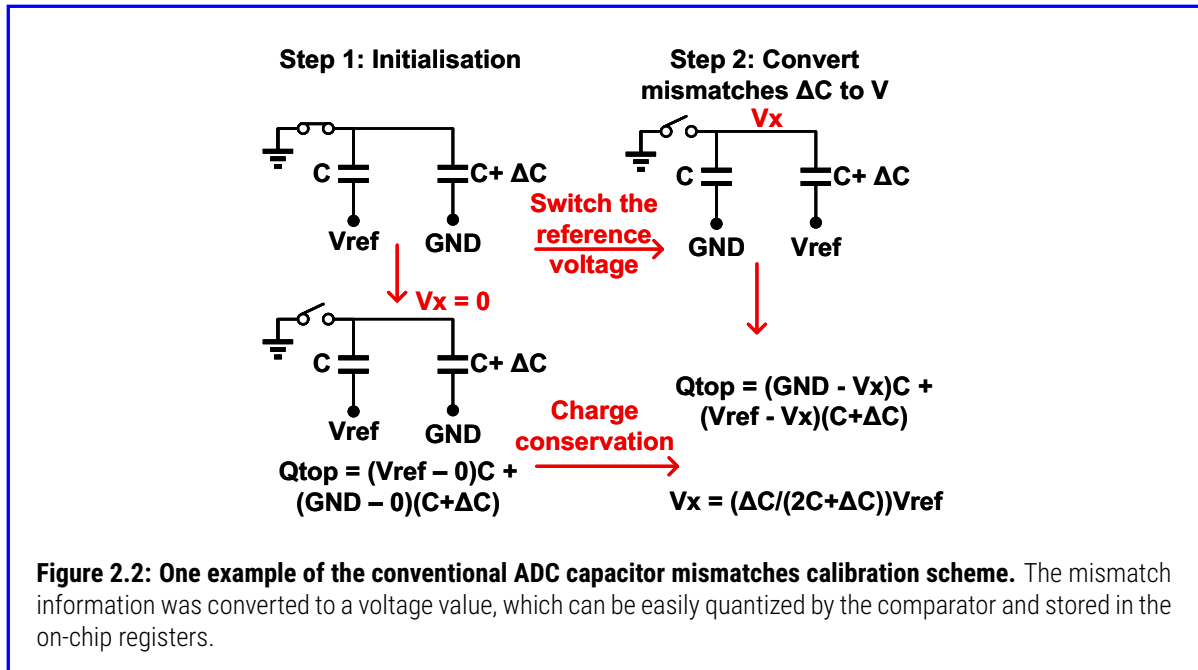
Modern ADCs often incorporate self-calibrating mechanism that automatically trim for errors without external intervention. These techniques include both analog and digital calibration methods. Recent advancements from the ADC survey [9] have demonstrates various calibration techniques to improve the overall ADC performance, including exploiting the emerging machine learning algorithms to predict the ADC non-linearities. These methods can potentially offer higher accuracy and adaptability compared to conventional calibration techniques. However, they are still in the experimental stage and require further research to be widely adopted by the industry.

Despite substantial progress in calibration methodologies, fundamental challenges persist in high-resolution ADC calibration. A primary constraint manifests in the inherent complexity and power consumption characteristics associated with high-resolution architectures. The implementation of sophisticated digital calibration techniques, including HDC and DNC, introduces significant overhead in power dissipation and architectural complexity. For example, as shown in Figure 2.2, the conventional way to measure the capacitor mismatches inside the DAC arrays is to simply switch the reference voltage between MSB and LSB. Then, the mismatches are transferred to a voltage value, which can be easily quantized by the comparator. However, for a high resolution ADC, this measuring progress needs to be repeated for every calibration bits, lowering the speed of overall conversion. The on-chip registers used to store the calibration information also cost extra area and power to the ADC system.

This challenge is compounded by the fundamental trade-off between calibration precision and operational speed, particularly in achieving fine-granularity calibration at the sub-LSB level. The pursuit of enhanced calibration accuracy frequently necessitates compromises in operational speed, representing a critical design consideration in ADC development. Continuous calibration approaches, while offering superior accuracy, often impose limitations on ADC operational frequency. The calibration of high-resolution ADCs encounters several critical bottlenecks that demand innovative solutions for performance and efficiency optimization. A fundamental limitation arises from the algorithmic complexity inherent in calibration methodologies. As ADC resolution requirements escalate, the corresponding calibration algorithms demand increasingly sophisticated computational resources, resulting in elevated power consumption profiles. This power overhead presents partic-

ular challenges in energy-constrained applications, such as battery-powered devices and portable systems [2, 9, 59].

The speed-accuracy trade-off emerges as another significant constraint in high-resolution ADC calibration. The requirement for precise calibration in high-resolution architectures inherently conflicts with operational speed considerations. Continuous calibration techniques, while capable of achieving calibration high accuracy, typically impose limitations on ADC operational frequency, potentially rendering them unsuitable for high-speed applications. The optimization of this fundamental trade-off between calibration precision and operational speed remains a critical challenge in high-resolution ADC design methodology. Success in addressing these constraints requires innovative approaches that can effectively balance the competing requirements of accuracy, speed, and power efficiency within the context of specific application requirements [2, 9, 59].



2.2.3 Conclusion

The comprehensive review of memristor technology and ADC calibration techniques reveals interesting opportunities at their intersection. While memristors have demonstrated promising capabilities in various compute-in-memory applications, their potential for enhancing mixed-signal circuits remains largely unexplored. Concurrently, ADC calibration techniques face persistent challenges in balancing performance improvements with overhead costs in power, area, and latency. The memristor's unique properties - particularly its non-volatility, resistance tunability, and compact size - could potentially address these fundamental limitations in ADC calibration.

The convergence of these two domains presents an opportunity to develop novel calibration approaches that leverage memristors' advantages while mitigating their current technological limitations. By integrating memristors into ADC calibration schemes, it may be possible to achieve more efficient implementations that reduce the conventional overhead associated with conventional calibration techniques. This approach could be particularly beneficial for array-based interface applications where power and area constraints are critical.

However, successful implementation of memristor-assisted ADC calibration requires careful consideration of current memristor technology limitations, including device variability, retention degradation, and modeling accuracy. The following chapters explore this novel approach, presenting circuit-level implementations that demonstrate the feasibility of using memristors to enhance ADC performance while addressing these technical challenges.

Memristor-assisted Background Calibration for a 12-bit SAR ADC with RDAC

This chapter presents a memristor-assisted background calibration scheme implemented in a 12-bit asynchronous successive approximation register analog-to-digital converter (SAR ADC), which consists of a hybrid binary weighted/R-2R digital-to-analog converter (binary/R-2R DAC) and other peripheral circuits. This hybrid DAC, in which one redundancy bit is introduced, is built with a memristor and standard polysilicon resistors. The proposed calibration technique can detect the errors caused by DAC mismatches and correct them by adjusting the resistance of the memristor in a feedback loop. The implemented circuit takes the memristor's advantages such as small area and resistance switching property. The proposed scheme has been designed and simulated in a standard 180 nm CMOS process. Eventually, a monolithic CMOS/memristor chip was fabricated with the CMOS part processed at a standard foundry and the memristors integrated through post-CMOS processing in house. The results of this research have demonstrated the feasibility of exploiting memristors to improve the linearity of high-resolution SAR ADCs. The designed calibration scheme can effectively reduce the integral non-linearity (INL) and differential non-linearity (DNL), so improve the signal-to-noise and distortion ratio (SNDR) by 14.26 dB and the spurious free dynamic range (SFDR) by 21.11 dB. To the best of the author's knowledge, this is the first published memristor-assisted calibration scheme for ADCs. Despite there are still many limitations at this stage, with the presented results, the author would like to draw attentions of the ADC design community to an alternative calibration scheme enabled by the emerging memristor technology.

This chapter was published in the **IEEE Transactions on Circuits and Systems I: Regular Papers (Vol: 70, Issue: 9, Sep. 2023)** [2], invited by **2022 20th IEEE Interregional NEWCAS Conference (NEWCAS)** [61]. The author's work covers all the system-level and transistor-level designs and simulations of the proposed ADC. The layout and chip test works are all done by the author. © 2025 IEEE.

3.1 Introduction

A memristor, whose concept was first proposed by Leon Chua in 1971, is a two-terminal device that represents the relationship between electric charge and magnetic flux [4]. In 2008, the first physical memristor device was realised in Hewlett-Packard (HP) Laboratory, and it exhibits a resistance switching property correlating to amplitude, frequency, and polarity of applied voltages [5]. Since then, the memristor has become the focus of many studies. Over the last decade, as CMOS downscaling is approaching its physical limits, there are higher demands on emerging compute-in-memory (CIM) devices which can overcome the conventional von Neumann architecture. The memristor is one of these CIM devices, and its potential has been proven in various applications such as non-volatile memories and in-memory computing, etc [27, 62–66]. Up to 2024, the memristor and other CIM devices have played an important role in improving the future computing power to assist artificial intelligence (AI) development (e.g., the CIM device has become one of the future national goals of the US government [1]). The capability of integrating the nanoscale memristors with the CMOS technology makes the memristor an ideal candidate in next-generation hybrid system-on-a-chip (SoC) designs. The memristor's properties such as small feature size down to 2 nm [6], resistance tuning, and non-volatility provide breakthroughs and potential solutions for traditional analog and digital circuit designs [67].

The performance of many CMOS circuits has benefited from advanced CMOS technology, and one example is the successive approximation (SAR) ADC. While SAR ADCs are energy efficient, their resolution is typically limited by element matching, which leads to mismatch and offset issues [59]. The mismatch issue can be mitigated by increasing the element size (i.e., unit capacitance (C_u), unit resistance (R_u)), but this will decrease the ADC's efficiency (i.e., chip area, power consumption, and latency). For instance, the area of the intrinsic DAC has to be scaled up four times for one extra effective bit to meet the matching requirements in capacitive SAR ADCs [9]. To deal with these limitations, various calibration techniques have been proposed to pursue higher resolution by trading off area, power, circuit complexity, and speed [59, 68–70].

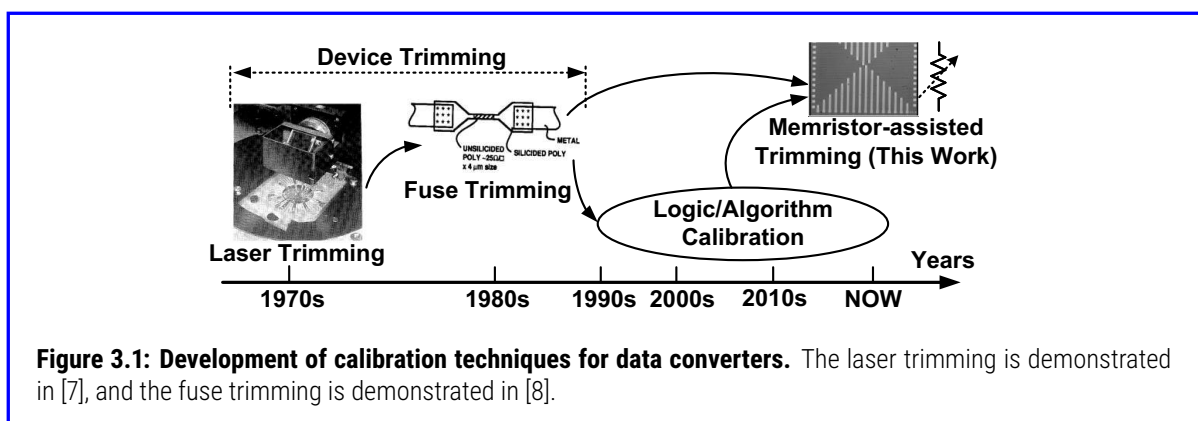


Figure 3.1 shows the development of existing calibration techniques over the past decades. Laser trimming and fuse trimming are the earliest calibration methods for data converters, and they both perform the calibration by changing the characteristics of the fundamental circuit elements (i.e., resistors) in an off-chip way. However, these methods suffered from different drawbacks. Laser trimming requires a laser and thin-film process, which raises the complexity of fabrication and packaging. The laser also suffers from drifting issue [7]. As for fuse trimming, probe pads often limit the efficacy of calibration. Although there are techniques to overcome this problem [8], fuse trimming is still an irreversible operation and brings difficulties to ADC users. Furthermore, both trimming methods increase the cost of the ADC design, and the trimming efficacy can be easily affected by the packaging. After the 1990s, with the development very-large-scale-integration (VLSI), most calibration methods are based on digital logic and algorithms. These calibration methods exploit the high-performance CMOS devices to extract the calibration information through the mathematical way and perform the on-chip calibration in a feedback loop (e.g., noise-shaping, digital post-processing and analog background calibration). Implementing these methods generally requires on-chip memory (i.e., registers) and other peripheral digital circuits (e.g., counters, adders), which thus introduce extra overheads on power, area and latency. Thanks to the CMOS down scaling over the past decade, the performance of SAR ADCs has been greatly improved with the assistance of these calibration methods. However, as the CMOS down scaling is approaching to its physical limits, it becomes difficult to improve the SAR ADC performance further [9], and the extra overheads spent on calibration limits the F.O.M of the SAR ADC.

Besides existing calibration techniques, the memristor has the potential to be adopted as a trimming device for analog adjustments as well as a memory device (i.e, CIM) that facilitates calibration logic and algorithms, which can reduce the calibration overheads. Notably, memristors can be integrated into the CMOS back end of line, and thus monolithic chips can be developed [71], making the memristor-assisted trimming a promising solution for ADC calibration.

In this chapter, a memristor-assisted sign-based background calibration scheme for SAR ADCs with RDAC has been proposed. This new calibration technique has been implemented in a 12-bit asynchronous SAR ADC with conventional 'trial & error' switching scheme and its effectiveness in improving ADC linearity and area efficacy have been validated. The proposed SAR ADC contains a 13-bit hybrid binary/R-2R DAC (1-bit redundancy), in which the memristor-based trimming is introduced. The complete circuitry is designed and simulated in a standard 180 nm Bipolar-CMOS-DMOS (BCD) process. The memristor model used for circuit simulations was developed and validated from a physically fabricated memristor in [39]. Through this memristor-assisted background calibration scheme, the errors caused by element mismatches in the DAC are mitigated and the linearity of the SAR ADC has been effectively improved.

3.2 The Concept of the Proposed Methodology

As a result of the nature of the SAR ADC, intrinsic element mismatches in the DAC are often the dominating factor that affects ADC performance such as integral non-linearity (INL), differential non-linearity (DNL), Spurious-Free Dynamic Range (SFDR), and Signal-to-Noise-and-Distortion Ratio (SNDR). To address the mismatch issue, different techniques have been proposed to calibrate the ADC. Figure 3.2 (a) shows an example of the sign-based calibration scheme proposed in [59]. Unlike post-processing the output of the ADC in [70], this sign-based calibration can run in parallel with the normal ADC conversion and calibrate the output immediately when there is an error detected. This calibration scheme is implemented with a sign-error detection circuit, calibration registers, and a trimming circuit (i.e., correction DAC). Once the calibration is activated, with the assistance of the redundancy cycle, one sign error (i.e., "+" or "-" or "0") is generated by the detection circuit and stored in calibration registers. This sign error is then exploited to indicate the direction of the calibration and activate the correction DAC to trim the main DAC. The output of the main DAC can be trimmed to the desired reference level after some specific capacitors inside the correction DAC are activated or deactivated. The advantage of this sign-based calibration scheme is that it runs in the background, so normal ADC conversions are not interrupted during the calibration process. Compared to the digital calibration techniques such as detecting the full-bit output of the ADC in the digital domain [70], this sign-based calibration technique operates in the analog domain and achieved 5.5 fJ per conversion step in Walden figure-of-merit (Walden F.O.M). This work is still competitive when compared to the recent state-of-the-art SAR ADC being designed in the same technology node with different calibration schemes such as [72].

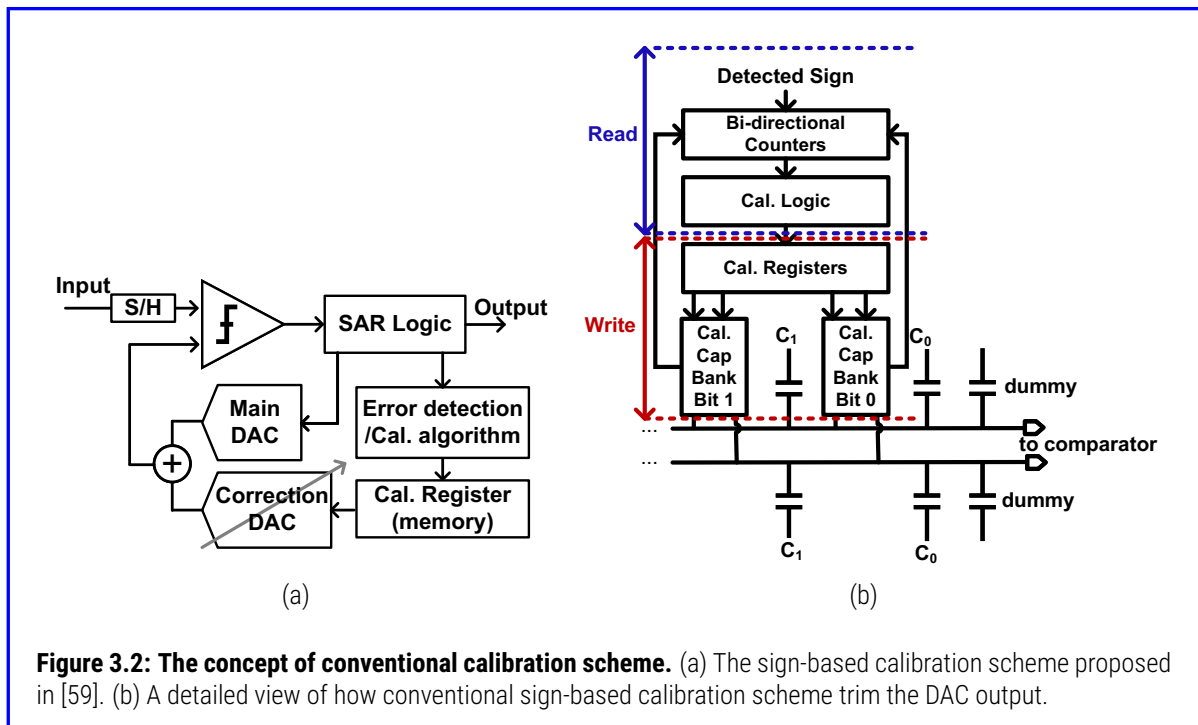
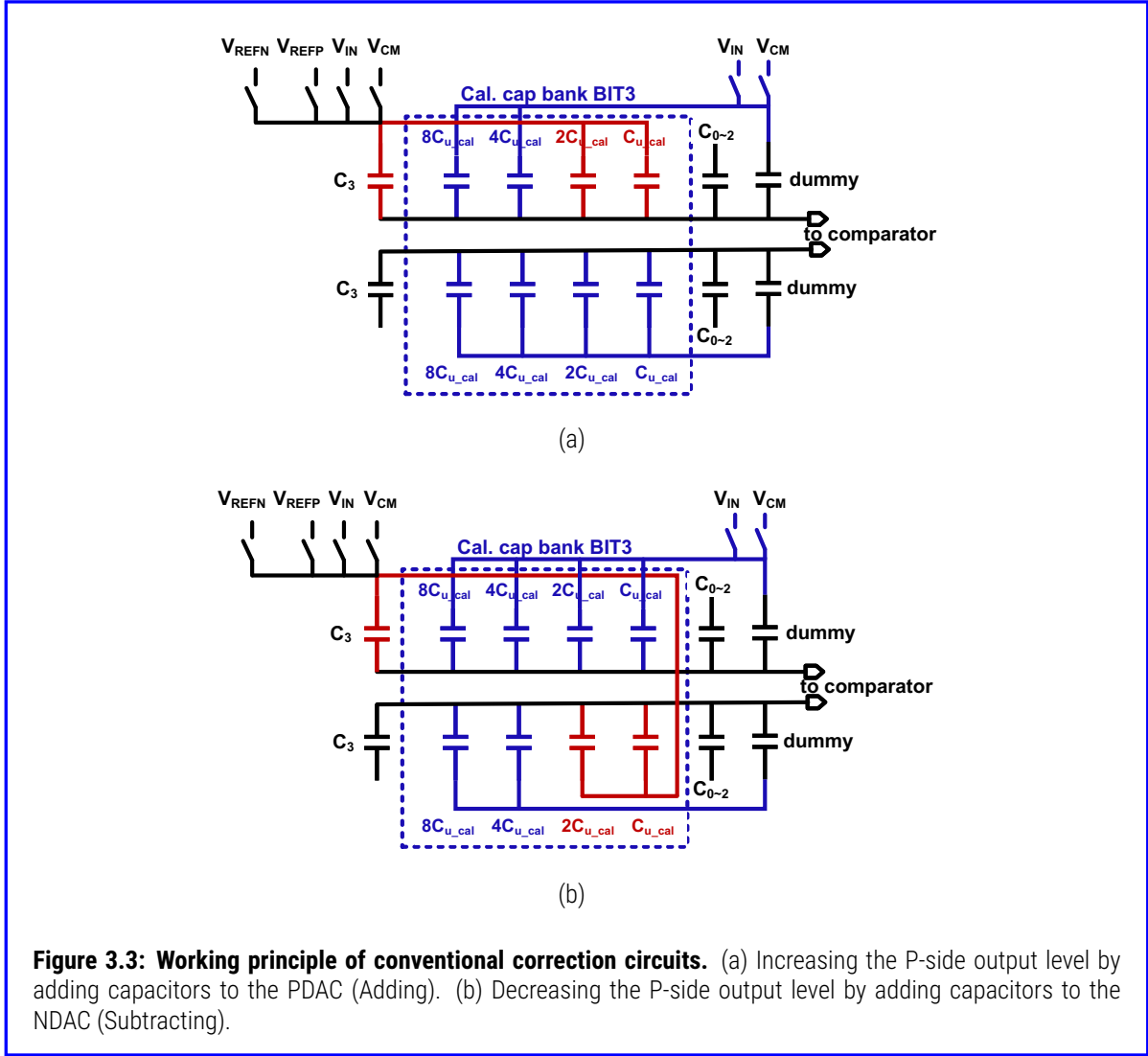


Figure 3.2: The concept of conventional calibration scheme. (a) The sign-based calibration scheme proposed in [59]. (b) A detailed view of how conventional sign-based calibration scheme trim the DAC output.



However, the conventional sign-based calibration scheme still takes considerable design overheads. Figure 3.2 (b) and 3.3 show a detailed view of how the conventional scheme trim the DAC output level. After detecting the calibration information (i.e., sign), the bi-directional counters choose to go up or down according to the sign. The counting number of the counters determine how many correction capacitors will be added (or subtracted) to (from) the main DAC bit. Then, the bi-directional counters can activate the registers which store the calibration information, and corresponding calibration capacitors are added to either P-side (adding) or N-side (subtracting) of the DAC according to the sign. The drawback of this conventional calibration scheme is that each calibration bit (e.g., from MSB to MSB - 4) needs its own calibration capacitor bank (correction DAC) and a counter to record the number of correction capacitors applied to the main DAC (for a 5 calibration bits example, the correction circuit includes 5 counters, 10 groups of capacitor banks and other logic circuits). This means a significant area occupation and power consumption, although the modern foundry can provide high-performance digital standard cells. Besides, it is obvious that the concept of the conventional calibration scheme is very similar to the von Neumann architec-

ture (e.g., the Cal. registers in Figure 3.2 (b) plays the same role as the cache memory playing in a digital computing system), which needs cooperation among individual memory units and other circuits to perform the read and write operation, increasing the system latency. Thus, the aim of this work is to optimise further the sign-based SAR ADC calibration scheme by combining the calibration registers and the correction elements into one single block with the assistance of the CIM memristor.

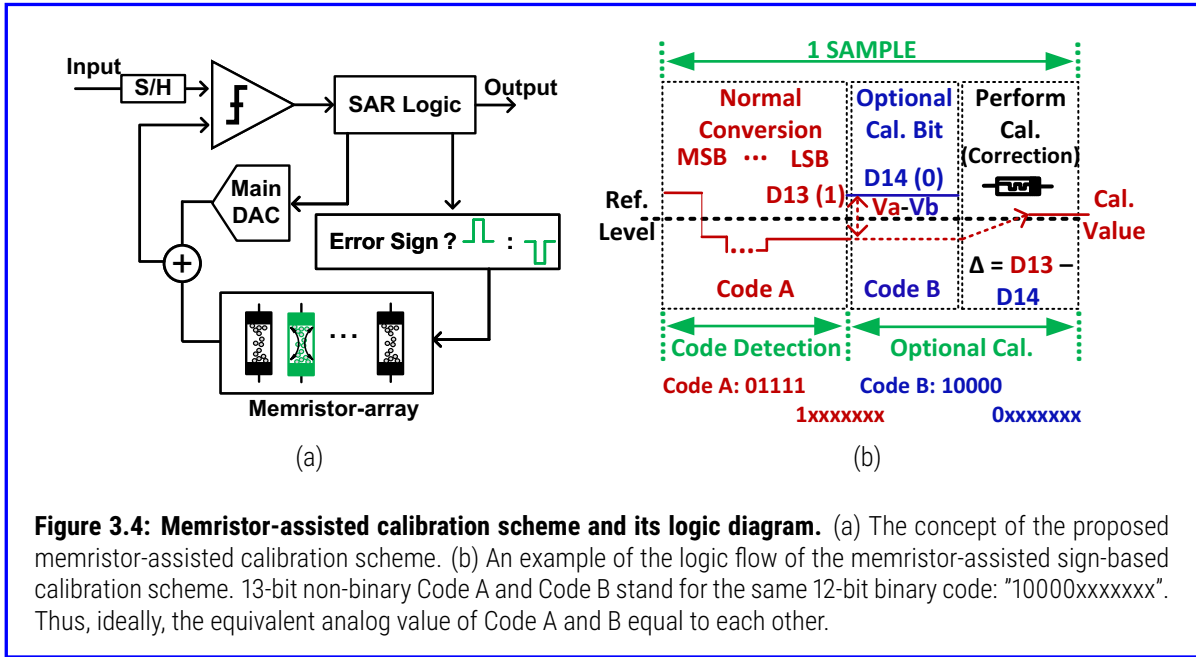


Figure 3.4: Memristor-assisted calibration scheme and its logic diagram. (a) The concept of the proposed memristor-assisted calibration scheme. (b) An example of the logic flow of the memristor-assisted sign-based calibration scheme. 13-bit non-binary Code A and Code B stand for the same 12-bit binary code: "10000xxxxxxx". Thus, ideally, the equivalent analog value of Code A and B equal to each other.

As shown in Figure 3.4 (a), thanks to the memristor's non-volatile resistive switching property, the proposed scheme is enabled by using memristors which can be used as resistive memory cells and resistive trimming elements at the same time. In a calibration process, the sign error works as a feedback signal which can determine the polarity of programming voltage pulses. The calibration can be easily conducted by applying positive or negative voltage pulses on the memristor according to the sign error. The memristance can thus be increased or decreased to make the DAC output reach the desired reference level. Compared to [59], non-volatility allows the memristor to store the calibration information and act as a trimming element at the same time, which can eliminate the excessive calibration registers (memories) and the correction DAC. In such a way, memristor-assisted calibration enhances the overall efficiency and further reduces the calibration latency and area occupation.

Figure 3.4 (b) shows an example of the logic flow of the proposed calibration scheme. The ADC first converts the input voltage in a 13-cycle normal conversion process, which includes 12 regular cycles and 1 redundant cycle. Thanks to redundancy, the same 12-bit code can be represented by two 13-bit non-binary codes (i.e., Code A and B in the figure). The DAC output of these two codes should be the same in ideal conditions (i.e., without matching issues). However, in reality, different DAC components (i.e., resistors in an RDAC, capacitors in an CDAC) suffer from different

mismatches, and the equivalent analog values of the two codes are not the same because of the intrinsic mismatch have different effects on the two codes. The calibration is activated if Code A or B is detected after the 13th conversion cycle. In the additional calibration (14th) cycle, one more conversion is performed. As shown in the example in Figure 3.4 (b), because of the difference between two DAC outputs, the conversion result D14 is "0", which is different from the 13th conversion result "1". This difference indicates a sign error (Δ). Based on this sign, in the following conversions, the memristance is adjusted to make the DAC output reach the desired reference level (i.e., correction), and the calibration is then finished. The calibrated DAC, which can provide more accurate D/A conversions, can benefit the subsequent A/D conversions and improve the output linearity of the ADC. More details of the working principle are described in Section 3.4.

3.3 The Available Memristor and its Model

Figure 3.5 (a) shows the micrograph of the in-house fabricated memristor used in this research. The characterization process was performed on a Pt/ Al_2O_3 / TiO_2 /Pt multi-state memristor using a customized probe station. A device response under stimulation of increasing amplitude pulse sequences is shown in Figure 3.5 (b). A set of 500 programming pulses with different amplitude V_b from 1.5 V to 2 V were applied to the memristor. 100 reading pulses were applied between the two adjacent stimulus pulse trains. The memristance¹ exhibits a gradual increase or decrease in response to the applied voltage of different amplitude and polarity, indicating that device behaviour can be controlled for calibration by incrementally or decrementally programming to different resistance states.

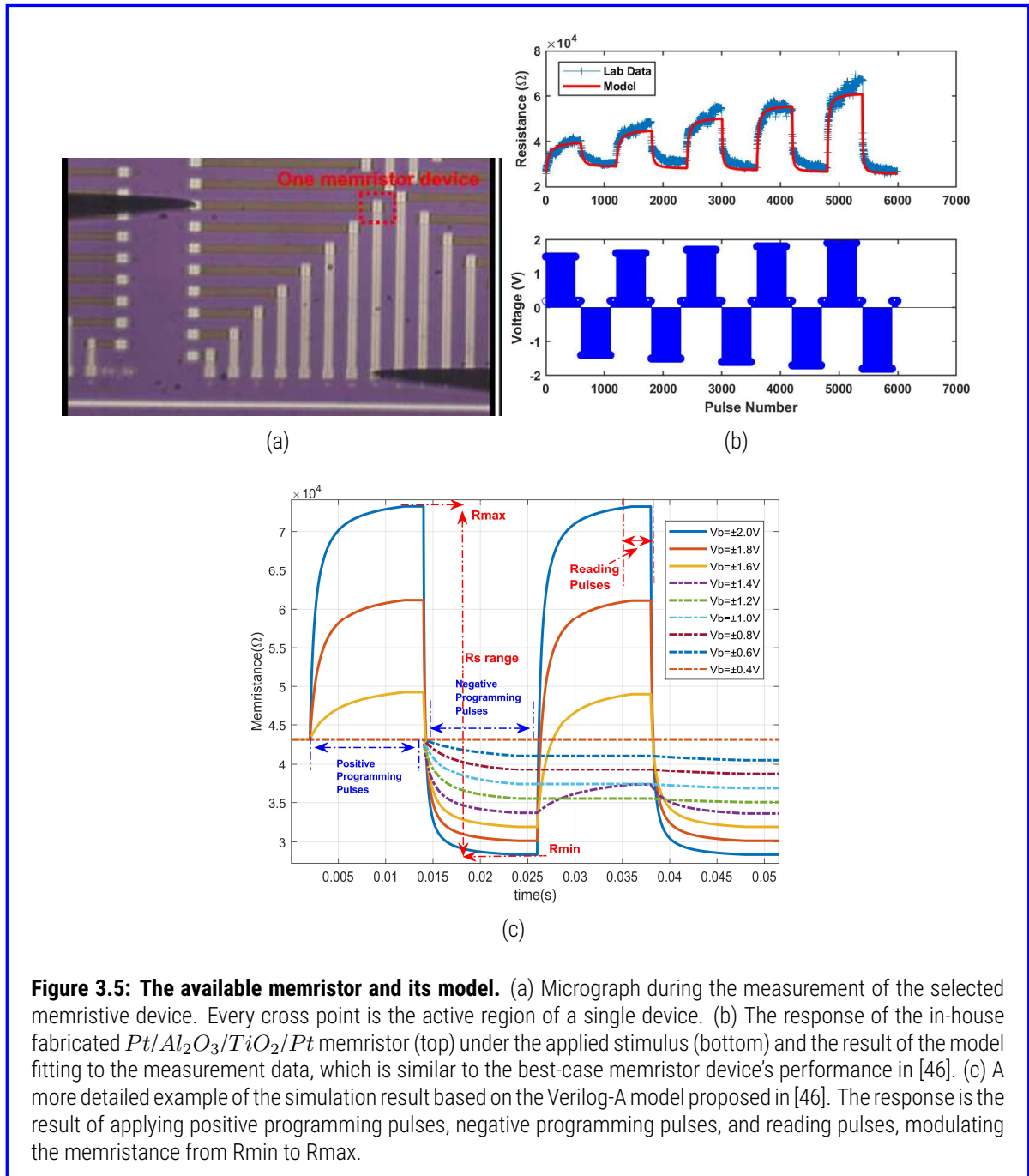
The memristor model used in simulations has been introduced and verified in [39, 46]. The static characteristics of the model consist of a bi-directional hyperbolic sine, whilst the rate-of-change of memristance exhibits an exponential dependence on the applied voltage. In particular, the switching rate surface is generated to reveal the reaction of the memristor to current resistance and applied voltage. The static current-voltage (IV) characteristic is decided by the device's internal state variable, Resistive State (RS), and a hyperbolic sine function of the voltage applied on the device's terminal. The time derivative of the state variable $\frac{dR}{dt}$ is decided by the sign and amplitude of V_b and RS value (i.e., $R < r_p(v), R \geq r_n(v)$ or $R > r_p(v), R \leq r_n(v)$). In Eq. 3.1, $s(v)$ serves as the switching sensitivity function, controlling the intensity of RS change. $(r(V_b) - R)^2$ corresponds to the window function which features a simple quadratic fitting and is regulated by the maximum/minimum RS $r(v)$ described in a linear form. $s(v)$ and $r(v)$ are defined in [39].

$$\frac{dR}{dt}|_{V_b} = \begin{cases} s_p(V_b)(r_p(V_b) - R)^2 \text{step}(r_p(V_b) - R) \\ s_n(V_b)(R - r_n(V_b))^2 \text{step}(R - r_n(V_b)) \end{cases} \quad (3.1)$$

As shown in Figure 3.5 (b) and (c), the model is fitted to the measurement data of the Pt/ Al_2O_3 / TiO_2 /Pt memristor and simulated with varied amplitude stimulation from ± 0.4 V to ± 2.0 V. Every programming cycle contains 500 pulses and the pulse width is 10 μ s. The simulated transient response demonstrates the same switching behaviour to the measurement results in Figure 3.5 (b). Initialized from $43 \text{ k}\Omega \pm 1 \text{ k}\Omega$, the memristor can be written up to $74 \text{ k}\Omega \pm 1 \text{ k}\Omega$ when applying positive voltage pulses, whereas negative voltage pulses can write the memristor down to $28 \text{ k}\Omega \pm 1 \text{ k}\Omega$. When the applied pulses have a maximum absolute voltage below 0.5 V, the device is prevented from switching and remains static. Thus, 0.5 V is set as the threshold volt-

¹The resistance of the memristor device is conventionally defined as 'memristance'. A stable memristance value is defined as a memristance state.

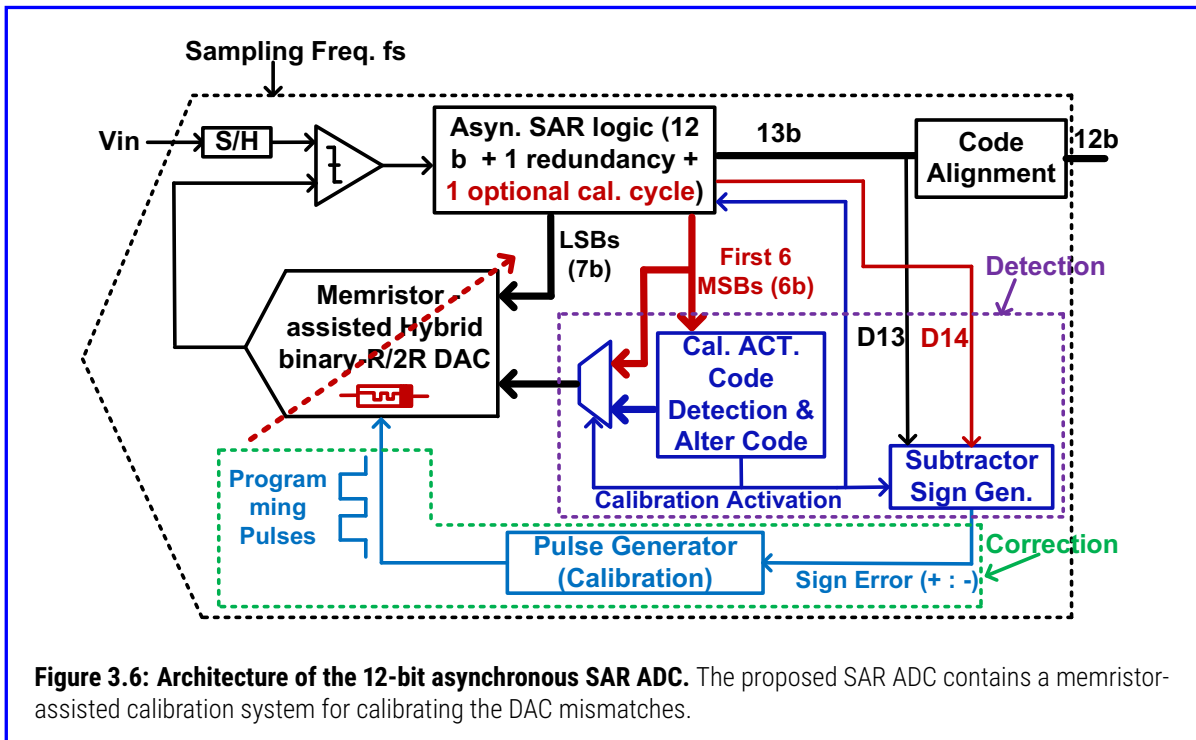
age in the design. It is worth mentioning that stimulus under $|1.4 \text{ V}|$ only changes the memristance by a small value, which raises a trade-off between power consumption, speed of calibration, and the covered range of calibration. Low voltage stimulus can be applied to the memristor when a fine calibration is needed, the trade-off can be balanced not only rely on the memristor but also on the technology node and the variations brought by the fabrication. In this design, a 2 V pulse train is used to trim the memristor so that the effect of calibration is maximized, this will be presented with more details in Section 3.5. The Verilog-A model can be easily adopted within standard CMOS design flow in Cadence. The physically extracted parameters used for the simulations can be found in [39].

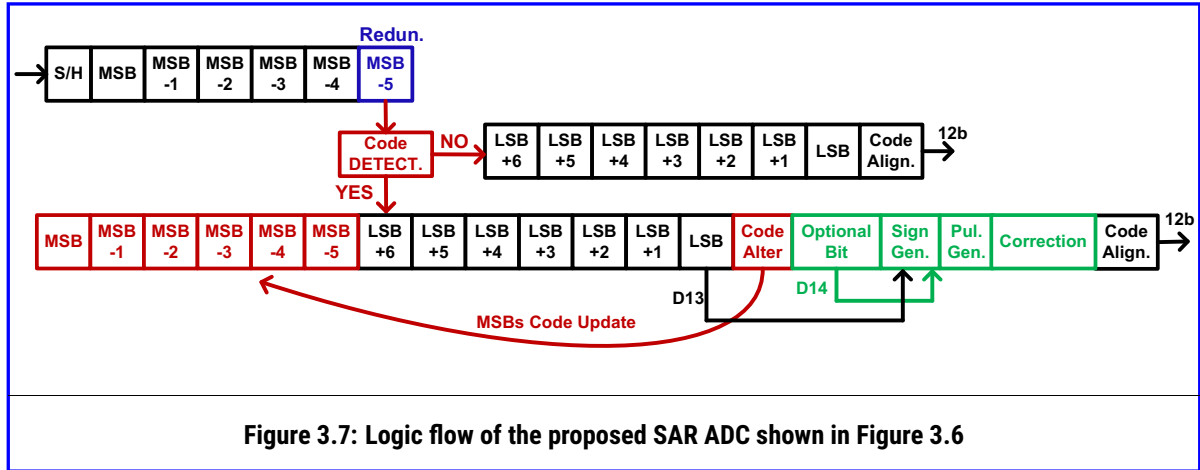


3.4 Implementation of the 12-bit SAR ADC with the Memristor-assisted Background Sign-based Calibration Scheme

3.4.1 SAR ADC Top-level Architecture

Figure 3.6 shows the proposed architecture of the ADC, which is a detailed block-level design developed from the scheme presented in Figure 3.4. The input voltage is sampled and held on a conventional S/H block proposed in [73]. The SAR logic includes 12 regular conversion cycles, 1 redundancy cycle, and 1 calibration cycle. The SAR logic is based on an asynchronous architecture assisted by TSPC logic, which is similar to the one proposed in [74]. This asynchronous logic is exploited to ensure that the calibration cycle can be successfully activated when one of the calibration activation codes is detected. This is because, the internal clock signals are generated by an internally triggered asynchronous logic, so each conversion cycle can only be triggered by the preceding cycle. Thus, the 14th calibration cycle can only be activated after the 13th conversion cycle is finished and the “Calibration Activation” signal is activated. The “Cal. ACT. Code Detection & Alter Code” and “Subtractor Sign Gen” circuits detect the “calibration activation codes” in Figure 3.15 and generate a feedback sign (i.e., detection). The “Pulse Generator” generates programming pulses based on the feedback sign to trim the memristor inside the DAC (i.e., correction). A detailed view of the proposed SAR ADC working logic flow is shown in Figure 3.7. The diagram shows how the internal logic works in the SAR ADC with and without the calibration activated.

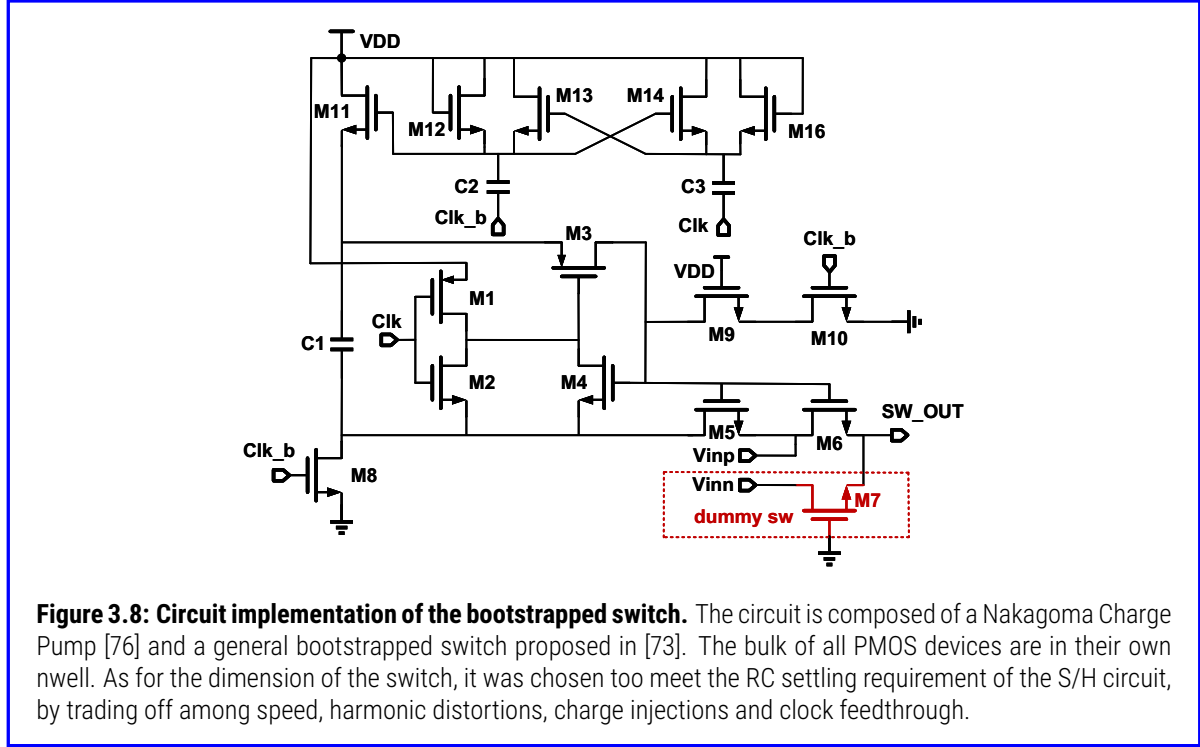




In a complete conversion of one sample, the first 6 MSBs are sent to a 6-bit multiplexer (MUX), which consists of six 2-to-1 MUXs and a calibration activation code detection circuit. When the 6 MSBs do not match any “calibration activation codes” in Figure 3.15, the 6 MSBs will not activate the detection circuits shown in Figure 3.15, thus the “Calibration Activation” signal keeps low, and the “Alter Code” circuit shown in Figure 3.6 will not be activated. The MUX then sends the original 6 MSBs to the DAC. In the DAC, the MSBs and the LSBs are recombined together into a 13-bit code. When the 6 MSBs match any “calibration activation codes” in Figure 3.15, the 6 MSBs activate the detection circuits, and the detection circuits then turn the “Calibration Activation” signal on, which activates the additional calibration cycle and the “Alter Code” circuit. Before the additional cycle, the altered 6 MSBs are sent to the DAC by the MUX, so the recombined 13-bit code is updated to its relative form (i.e., Code A to B or Code B to A). In the additional cycle, the comparator performs one extra comparison and determines the 14th compared result (D14). If the comparison result in the 13th cycle (D13) equals D14, the pulse generator will not be turned on and the memristor will not be trimmed. If D13 is not equal to D14, a subtractor logic will determine the sign of D13 minus D14 and generate a feedback signal. Next, the corresponding voltage pulses are generated to adjust the resistance of the memristor in the hybrid DAC. When the conversion is finished, the code alignment block converts the 13-bit non-binary code to a 12-bit binary code. This code alignment block is customized for the applied redundancy technique whose theory is first proposed in [75]. A detailed timing diagram that illustrates the above process is presented later in Section 3.5.

3.4.2 Implementation of the Bootstrapped Switch

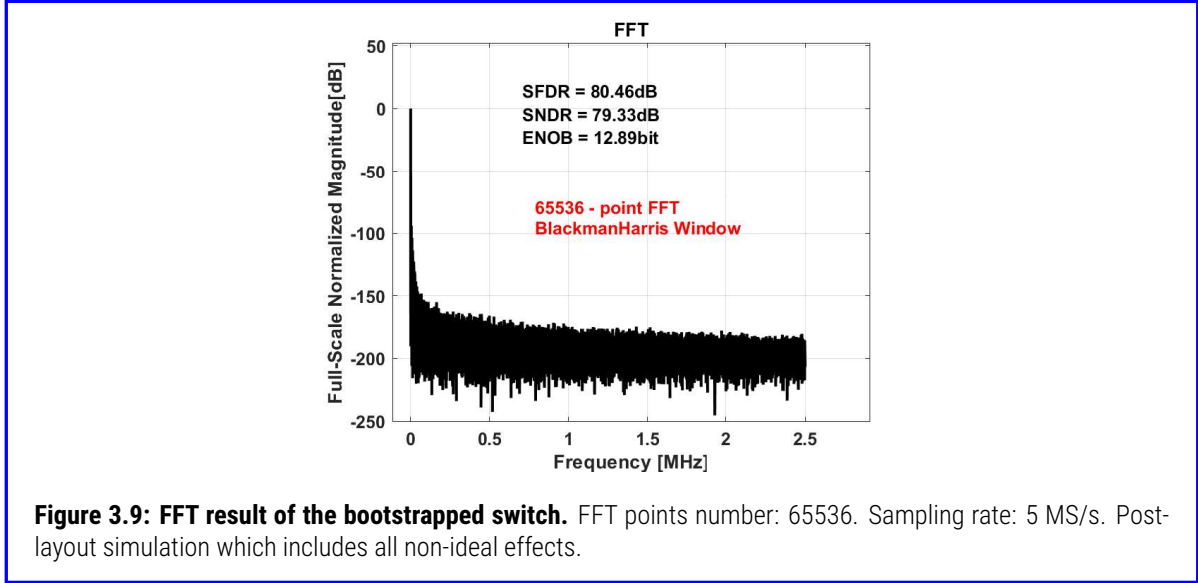
The input sampling switch plays a crucial role in A/D conversion as its linearity directly influence the ENOB of the ADC. A good sampling switch must have enough settling time to ensure the settling error ($\geq 9.01 \frac{T_s/2}{\tau}$ for this design)² is less than 0.5 LSB and a better linearity than the ADC. In this design, a bootstrapped switch [73] was used to maintain a good input linearity. The bootstrapped switch can ensure a stable V_{GS} voltage across the transistor M6 and thus a stable on-resistance, which is the conventional transmission gate (TR) switch cannot achieve.



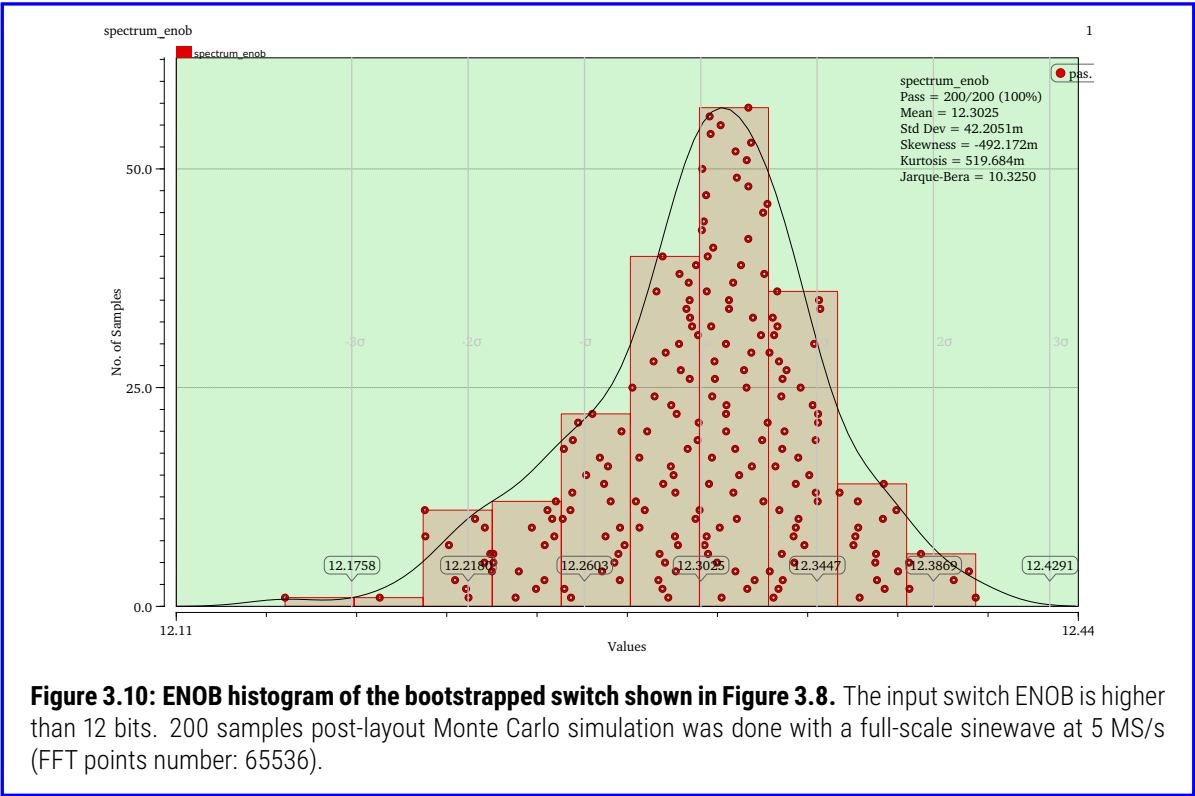
The value of C1 capacitance should be large enough to ensure a stable V_{GS} during the whole sampling period. In this design, the C1 was selected as 1.8 pF. The requirement for C2 and C3 are less tricky as they are part of the charge pumps to store a 2VDD reference voltage, and their values were selected as 100 fF. The dummy transistor M7, which has the same size as M6, is added here to reduce the coupling effects (from C_{ds} of M6) due to the variation of the input [75]. All capacitors used in this design are Metal-Insulator-Metal (MIM) capacitors because in the standard 180 nm process, the Metal-Oxide-Metal (MOM) capacitor does not has advantage on area occupation compared with the MIM for large capacitance value (The MOM capacitor is used in the bootstrapped switch design proposed in Chapter 4 because of its compact size in 65 nm process). Figure 3.9 shows the FFT result of the proposed switch at 5 MS/s and with 65536 (2^{16}) sample points. The proposed switch can achieve an ENOB of 12.89 bits, which is enough for the proposed 12-bit SAR ADC. Figure 3.10 shows the histogram of ENOB of the proposed bootstrapped switch, which is generated

²The first order RC response equation is $V_{out}(t) = V_{FS}(1 - e^{-t/\tau})$, then the settling error is $V_{out,err}(t) = -V_{FS}e^{-t/\tau}$. Assume the settling error needs to be smaller than 0.5 LSB, we can get a equation $\frac{T_s/2}{\tau} \geq \ln(2 \cdot 2^{ENOB})$.

from a 200-sample post-layout Monte Carlo simulation at a sample rate of 5 MS/s. These results prove that the proposed switch is able to maintain a good input linearity for the proposed ADC design. (N.B., the frequency of the input stimulus is 992 Hz, which is selected to achieve a coherent sampling. This means an integer number of cycles inside the predefined sampling window [77]. In this case, f_{sample} is 5 MHz, $N_{window} = 13$, $N_{record} = 65536$. Thus, the $f_{In} = \frac{5MHz \cdot 13}{65536}$.)



To calculate the time constant ($\tau = RC$) of the proposed RC network, the testbench shown in Appendix figure 5.1 was used to get the on-resistance (R_{on}). The holding capacitor used in this design is 1 pF, then the $T_{required} \geq \ln(2 \cdot 2^{12}) \cdot 2R_{on}C = 6.8ns$. The sampling rate of the design is far more enough for the sampled signal to settle ($1/5 \text{ MHz} = 200 \text{ ns}$). The 1 pF holding capacitor can also easily meets the thermal noise requirement of the sampling network, i.e., $\frac{kT}{C} \leq \frac{V_{LSB}^2}{12}$.



3.4.3 Implementation of the Comparator

The comparator used in this design is in a dynamic architecture, which includes a pre-amplification stage, a regenerative latch stage and an output buffer (SR latch). By exploiting the parasitic capacitance at nodes Vp and Vn, one side of the regenerative latch is going to be pulled up to VDD ('1'), while the other side will be forced to be GND ('0'). The output buffer is a SR latch which is used to generate a well-behaved digital output digit.

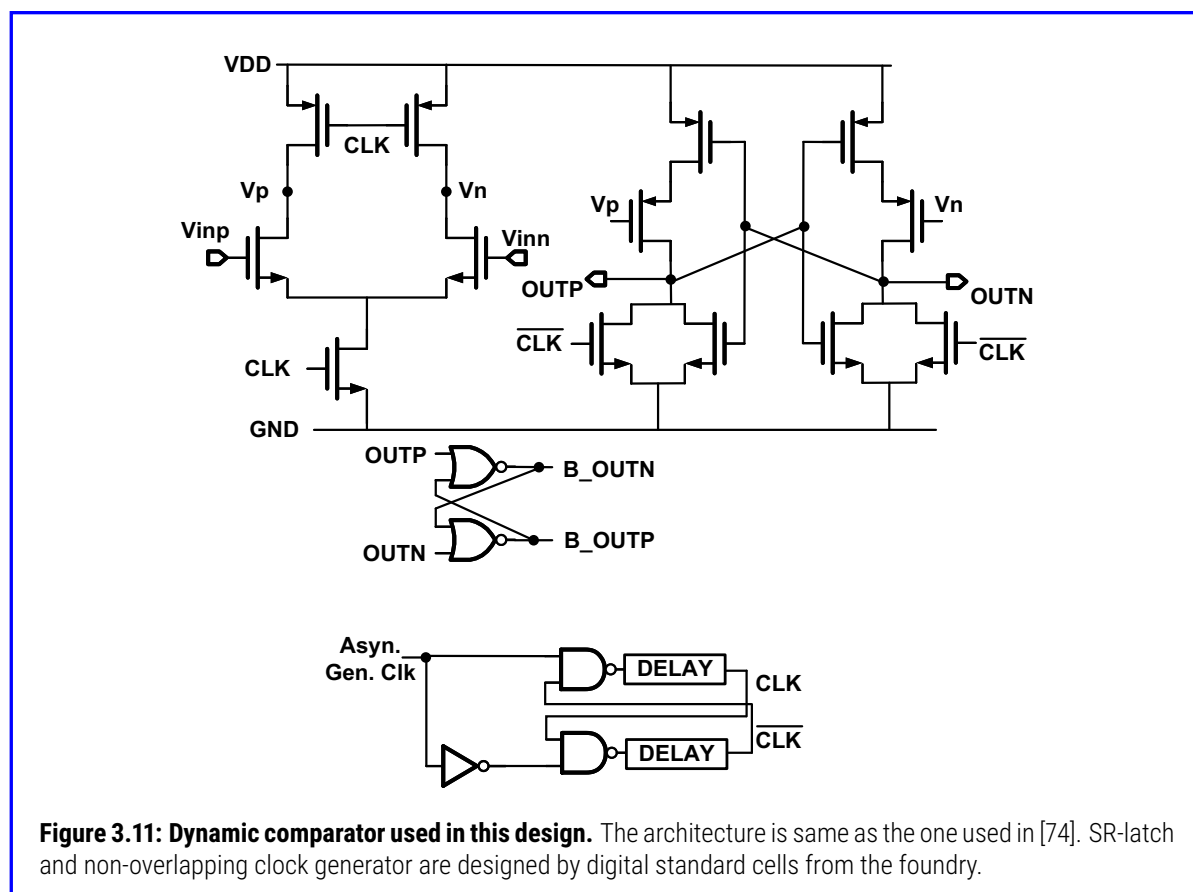
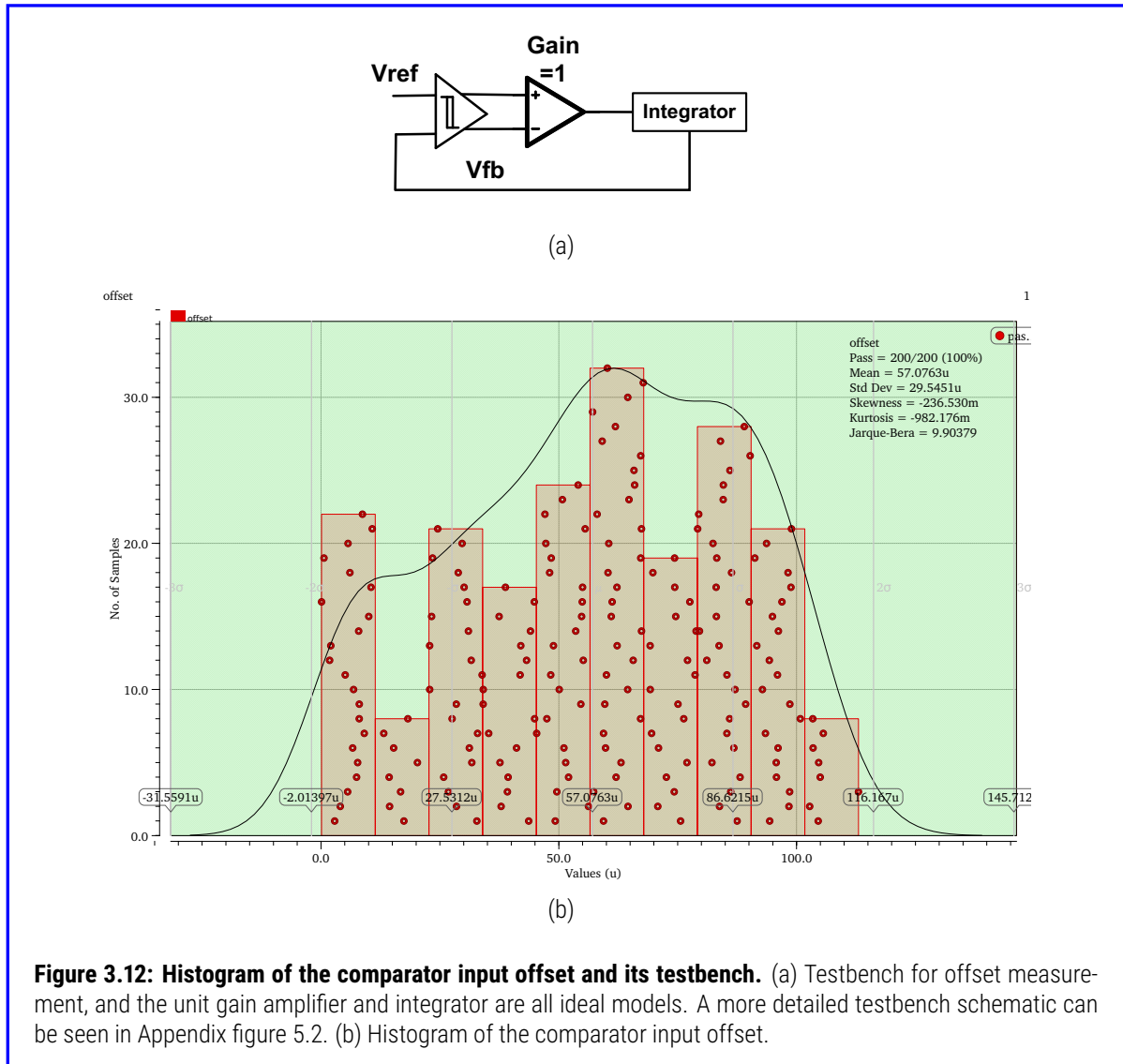


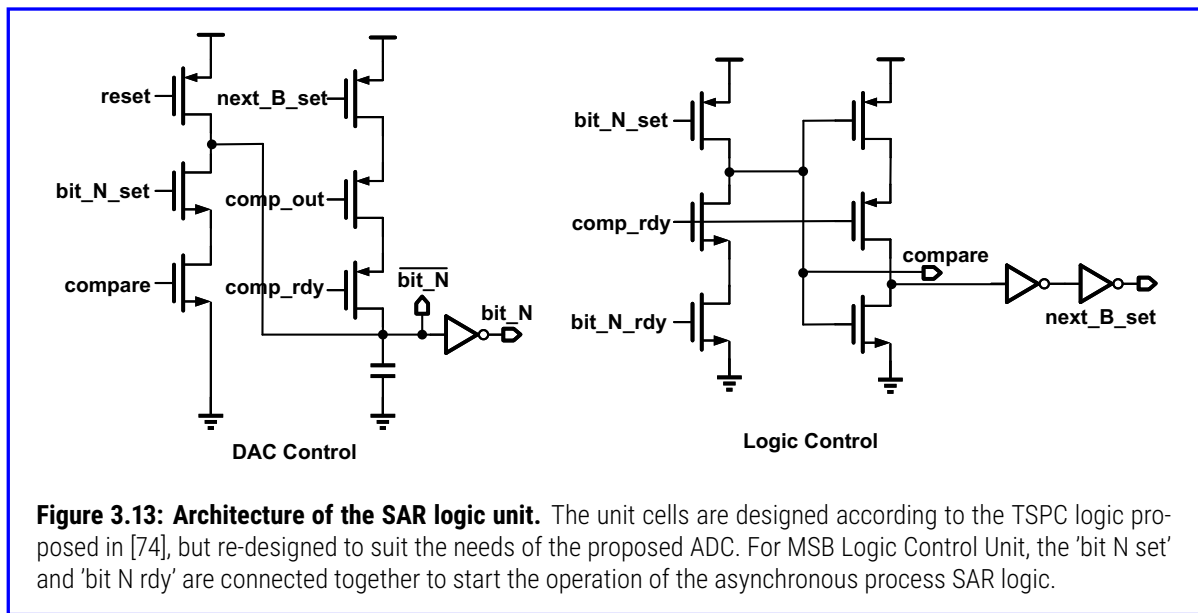
Figure 3.11: Dynamic comparator used in this design. The architecture is same as the one used in [74]. SR-latch and non-overlapping clock generator are designed by digital standard cells from the foundry.

Figure 3.12 shows the testbench to quantise the comparator input offset and the offset histogram got from a 200-sample Monte Carlo Simulation. To meet the design requirement of a 12-bit SAR ADC, the input offset is expected to be less than 0.5 LSB (i.e., $0.5V_{ref}/2^{12} = 0.22$ mV). From the Figure 3.12 (b), the proposed comparator is able to maintain an offset less than 120 μ V, so can meet the design requirement.



3.4.4 Implementation of the Asynchronous SAR Logic & Clock Generation

Conventional DFFs and TSPC-based logics are two most-common-use unit cells to achieve an asynchronous SAR logic. As the focus of this design is primarily on medium speed and low-power consumption, the TSPC logic is used as the unit cell of the SAR logic in this design (N.B. TSPC logic is good for its fast operation speed, low power consumption and small size, but is more sensitive to the PVT variation than DFFs. In this design, due to the relatively long programming time required by the memristor in each sampling period, a faster logic is preferred.). As illustrated in Figure 3.13, the TSPC logic employed in this design is based on the logic used in [74], with minor modifications to meet the specific requirements of this design. The logic control cell and the DAC control cell are able to operate together to achieve the conventional 'trial & error' switching scheme (i.e., in one conversion cycle, the sar logic will first set the bit to '1' (trial). If the comparator output is '1', the sar logic will keep the trial result, or the bit will be corrected to '0' (error).). Figure 3.14 shows the overall architecture of the SAR logic and the asynchronous clock generation circuitry.



As shown in Figure 3.14 (b), the logic control cells and DAC control cells of different bits are connected together. The 'trial' of one bit will only be triggered to '1' when its preceding bit is defined properly (i.e., preceding 'trial & error' operation has been finished.). The comparator output is connected to DAC control circuits to do the 'error' operation.

Figure 3.14 (b) shows the asynchronous clock generation control circuit, where asynchronous clock cycles are generated sequentially by exploiting two outputs from the dynamic comparator. When the comparator clock is low (reset), the NMOS transistor in the latch stage will pull two comparator offsets down to '0' ($\overline{OUTP}$ and $\overline{OUTN}$ are reset to '1'). In this state, the output of the second AND gate is '1'. Note that the third input of the first AND gate is set to '1' during the reset phase of the ADC. The PMOS transistor is turned on to charge its drain node to '1', and this state is maintained

until all conversion cycles are completed. Consequently, the output of the first AND gate remains at '1' until signal 'conv_finished' becomes '1'. After the comparator completes its first comparison, $\overline{OUTP}$ and $\overline{OUTN}$ become different, pulling the second AND gate output to '0' and generating a negative clock edge. The comparator then resets its two outputs, causing the AND gate output to return to '1', thereby generating a clock phase. In this way, the comparator is able to finish all the conversion cycles with the assistance of the clock generation control circuit. A delay circuit from the digital standard cell library is added at the second AND gate output to control the pulse width of the asynchronous generated clocks. After 13th cycle (without calibration) or 14th cycle (with calibration), the 'conv_finished' becomes '1' to stop the clock generation. In the case when calibration is not activated, the 'conv_finished' will be triggered by the bit_13_set signal. If the calibration is activated, 'conv_finished' will be triggered by bit_14_set signal and one extra clock phase will be generated. A more detailed timing logic diagram is shown in later Section 3.5.

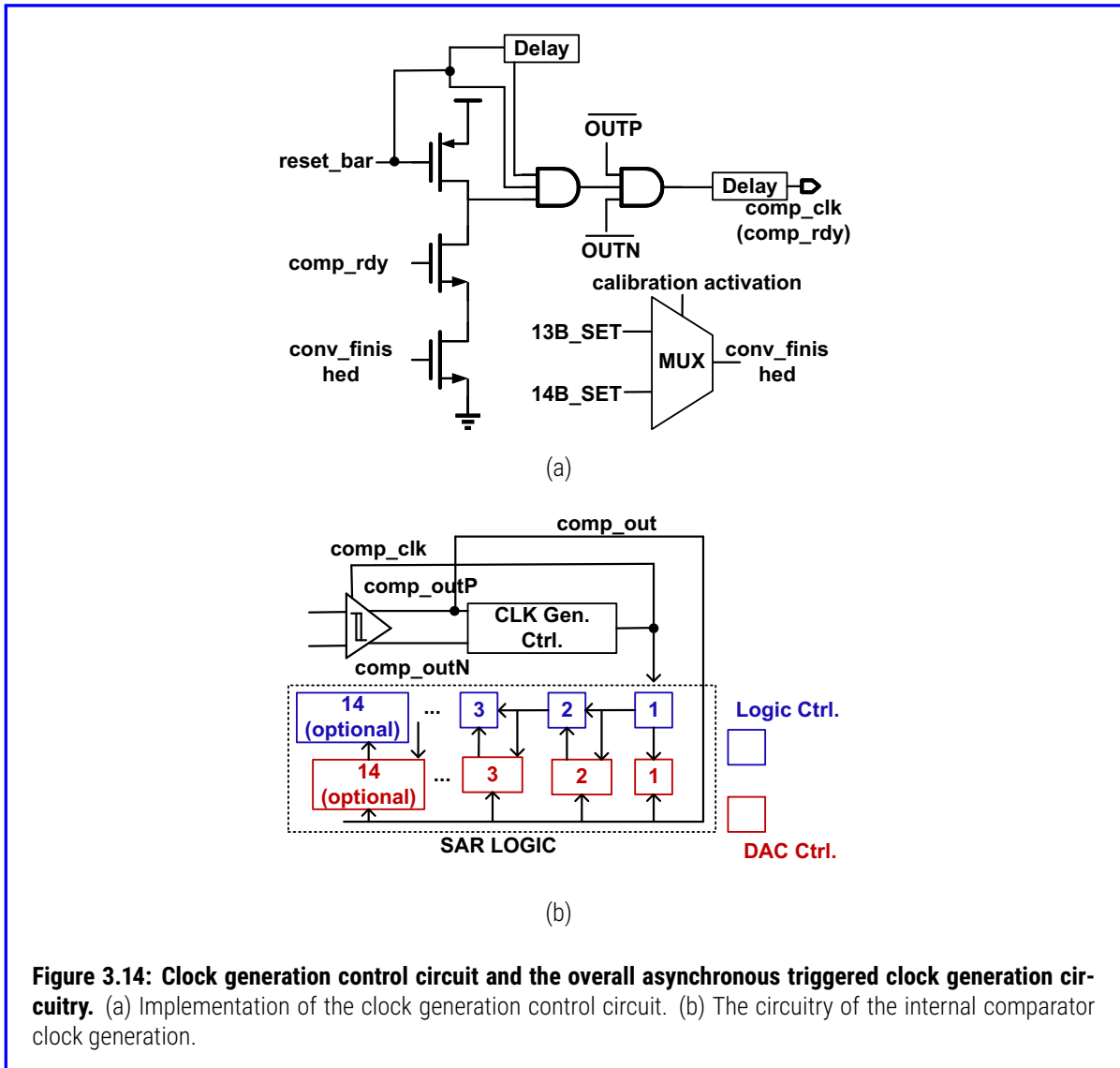


Figure 3.14: Clock generation control circuit and the overall asynchronous triggered clock generation circuitry. (a) Implementation of the clock generation control circuit. (b) The circuitry of the internal comparator clock generation.

3.4.5 Implementation of the Detection & Code Alteration Circuit

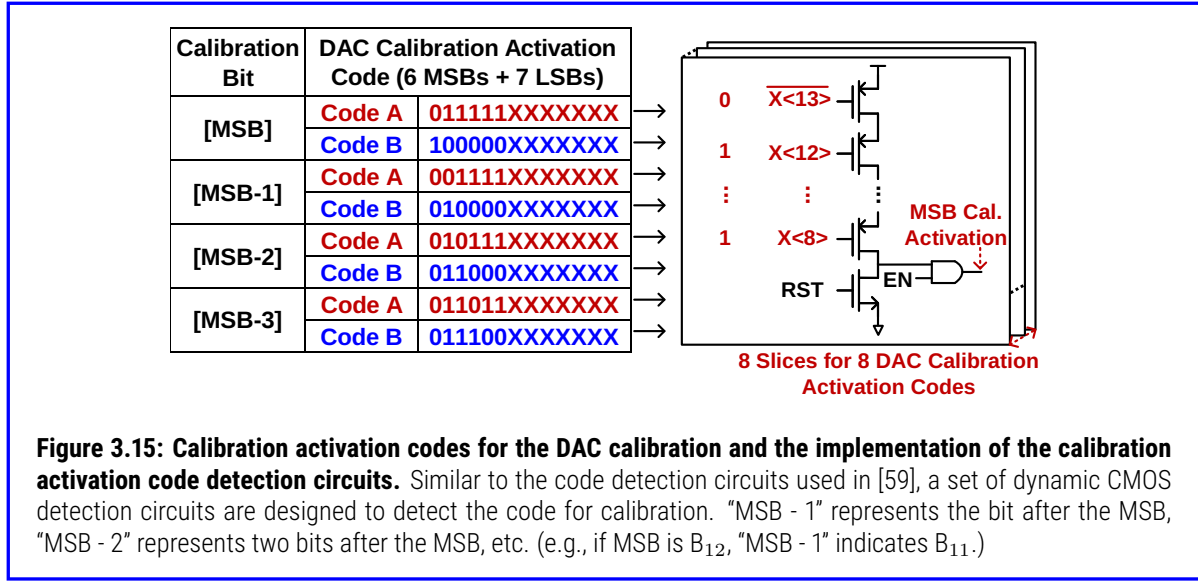


Figure 3.15 shows the calibration code detection logic. To avoid static power consumption, 8 specific codes can trigger the detection circuitry and turns on the “Calibration Activation” signal. Therefore, 8 slices of the detection circuit are implemented. Each detection circuit consists of 6 detecting transistors and 1 reset transistor. The 6 detecting transistors in each slice are customized to correspond to one of the 6-bit MSBs shown in Figure 3.15. To avoid unnecessary power dissipation and ensure there is enough time to activate the optional calibration cycle, the enable signal turns high at the 11th cycle in each conversion process. At the 11th cycle, if the 6 MSBs fit in one of the 8 scenarios, the corresponding calibration signal is charged to logic high and the additional calibration cycle is activated. Using code “011111” as an example, the 6 MSBs “011111” are sent to the detection circuitry and compared to the 8 scenarios at the 11th cycle. As 6 detecting transistors are all turned on, the “MSB Cal. Activation” signal is then charged to logic high, and the additional 14th cycle is activated. In this work, to minimize the unexpected non-ideal factors and further investigate the performance of a single on-chip memristor, only the MSB calibration is validated and 2 slices in code detection are used. In the future, trimming more MSBs, which may further improve the calibration performance, and thus the other 6 slices are also designed and implemented.

Figure 3.16 shows the implementation of the code alteration circuit. The 6 MSBs are sent to a set of half adders and half subtractors at the same time, as shown in the top and bottom of Figure 3.16. The last digit of the 6-bit MSBs determines whether the selection MUXs in the middle of Figure 3.16 select the codes from either the half adders or the half subtractors. This algorithm ensures that Code B can be generated if the input is Code A and vice versa. Still taking “011111” as an example. The last digit of this code is “1”, and therefore it is categorized as Code A. When this code is sent to the code alteration circuitry, “011111” turns to “100000” by the half adders and “011110” by the half subtractors. Because the last digit of the input code is “1”, the selection MUXs select “100000” as the output and thereby perform the code alteration. The same trick is applied to all 8 scenarios to

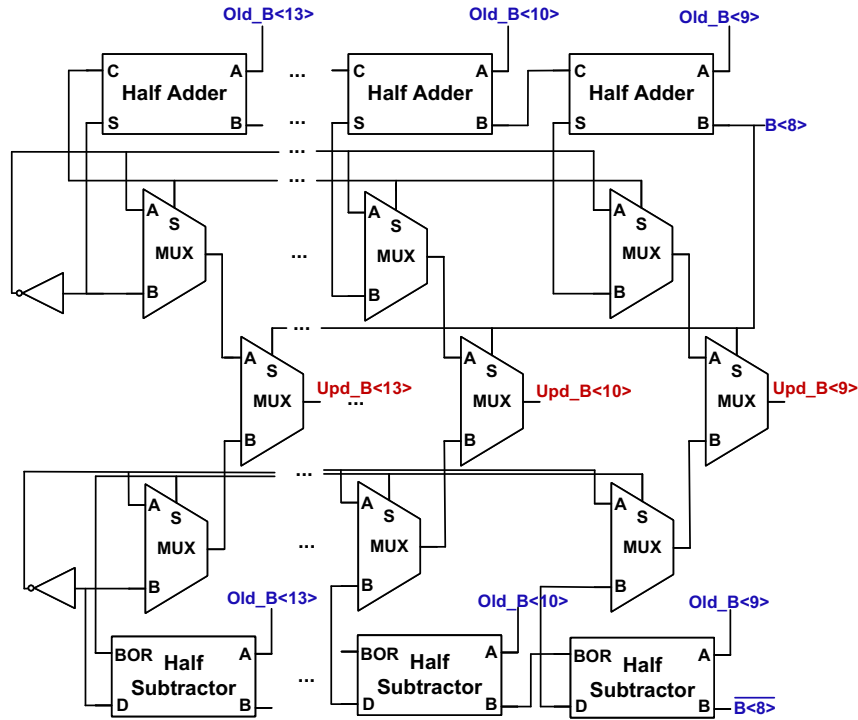


Figure 3.16: Implementation of the code alteration circuitry. Code A and Code B can be mutually altered by simply performing addition or subtraction in the digital domain. Half adders and multiplexers are from the standard cell library, and the half subtractor can be easily realised by standard AND, NOT, and XOR gates.

ensure Code A and Code B can be altered between each other. The equivalent analog value of the altered code is compared with the reference level in the calibration cycle and generates the 14th digit (D14). The D14 is then sent to the sign generator.

3.4.6 Implementation of Sign Generator and Pulse Generator

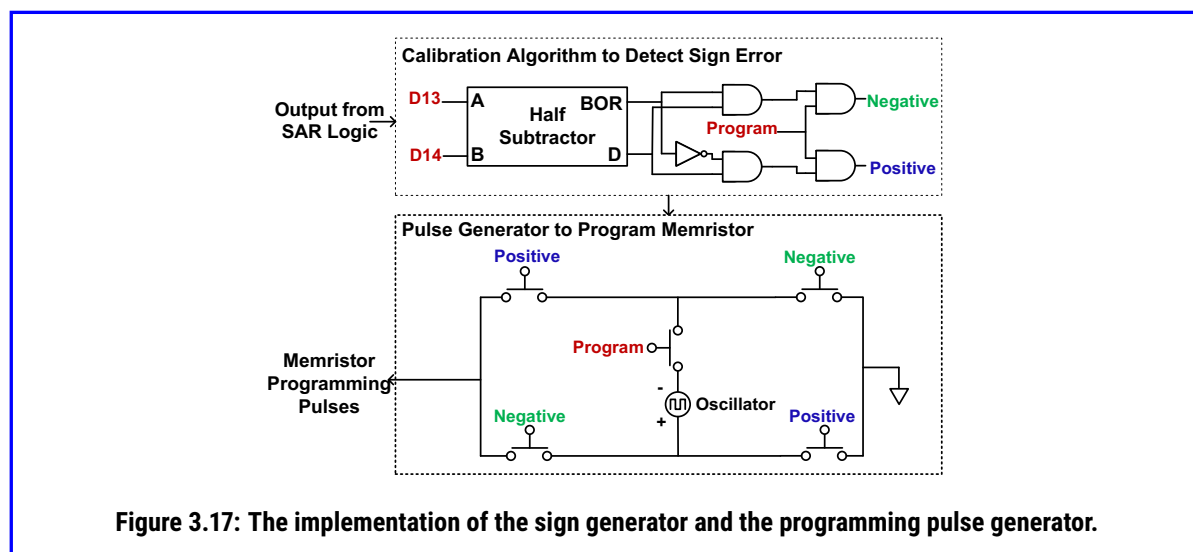


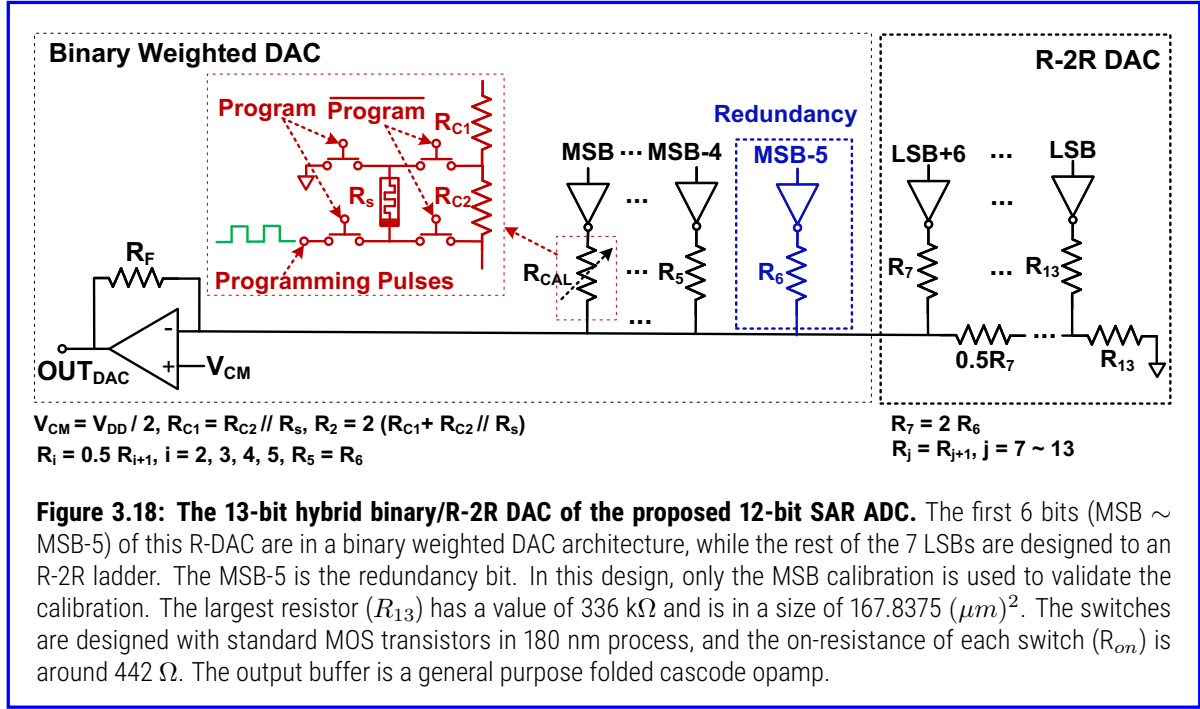
Figure 3.17 shows the sign generator and the pulse generator. The sign generator is designed with a half subtractor and five logic gates. When the calibration is activated, D14 is generated and sent to the half subtractor. The subtractor subtracts D14 from D13. When D14 and D13 have different values, the logic gates determine whether a positive or a negative signal is sent to the pulse generator. In the sign generator, the “Program” signal is the “Calibration Activation” signal with certain delay, and the pulse generator is turned on only when the “Program” signal is high. When D13 and D14 have equal values, the pulse generator is not turned on and no pulses will be generated. Table 3.1 shows the truth table of the sign generator and pulse generator circuits. “x” indicates no pulse is generated, “-” indicates negative pulses, and “+” indicates positive pulses. Note that when “Positive” is high, it means that the value of the MSB is lower than it should be, and therefore negative pulses are generated to trim the memristance to a lower value, which will increase the value of the MSB.

The pulse generator is designed based on an H bridge. When “Program” and “Positive” signals are received, positive pulses are generated and sent to the DAC to trim the memristor. The same process is performed for negative signals. To meet our design needs, the amplitude, pulse number, and pulse width are set to fixed values. By doing this, firm step sizes for the calibration can be set (e.g., the DAC output can be adjusted by 0.2 LSB in each calibration). Then, if the amplitude of the pulse is increased without changing the number and inter-pulse time, the DAC output can be adjusted by a higher step size of 0.4 LSB and vice versa. The pulses are generated by an oscillator.

Table 3.1: Logic of the sign generator and pulse generator

D13	D14	Program	Positive	Negative	Pulses
0	0	0	0	0	x
1	0	0	0	0	x
0	1	0	0	0	x
1	1	0	0	0	x
0	0	1	0	0	x
1	0	1	1	0	-
0	1	1	0	1	+
1	1	1	0	0	x

3.4.7 Hybrid Binary/R-2R DAC



In order to exploit the memristor as a trimming element in the DAC, a resistive DAC (R-DAC) is a straightforward design choice. Due to the use of the redundancy technique, a binary-weighted architecture needs to be applied to achieve the redundancy bit. However, assigning the resistors with binary weighted values leads to high power consumption and large area occupation. For example, in this design, the 12-bit SAR ADC requires a 13-bit DAC which includes 1-bit redundancy, and the DAC output needs to meet a non-binary relationship shown in Eq.3.2.

$$D_{out} = 2048 \cdot B_{12} + 1024 \cdot B_{11} + 512 \cdot B_{10} + 256 \cdot B_9 + 128 \cdot B_8 + 128 \cdot B_7(\text{Redun.}) + 64 \cdot B_6 + 32 \cdot B_5 + 16 \cdot B_4 + 8 \cdot B_3 + 4 \cdot B_2 + 2 \cdot B_1 + 1 \cdot B_0 \quad (3.2)$$

B_7 is the redundant bit, and the resistor value for this bit is the same as the one assigned to B_8 . If this R-DAC is fully in a binary weighted DAC architecture when the MSB resistor is 10.5 k Ω , the LSB resistor has to be 21.5 M Ω , and the total resistance is 43.165 M Ω . To reduce the overhead in power and area, a hybrid binary/R-2R DAC for the proposed ADC was designed.

As shown in Figure 3.18, the 7-bit LSBs are designed in an R-2R ladder architecture and the unit resistance is set to 336 k Ω ³. Because $R_7 = 2R_6$, the Thévenin-equivalent resistance of the R-2R ladder can restore a binary weighted relationship along with the MSB resistors to make this hybrid design fully functional. The MSBs are designed in a binary-weighted DAC architecture. The MSB

³The unit resistance is chosen to first meet the thermal noise requirement, which is $4kTR_{tot} < V_{LSB}^2/12$. Thus, the minimum R_{unit} can be confirmed. The second step is to run the matching test in the monte carlo simulation, the value here is a relative good matching value for the R_{unit} .

resistor R_{CAL} consists of a memristor labelled as R_S , and two resistors R_{C1} and R_{C2} . The R_{CAL} is formed by connecting the resistors R_{C1} and $R_{C2} // R_S$ in series. The equivalent resistance of R_S is 10.5 k Ω , where R_{C1} and $R_{C2} // R_S$ equal to 5.25 k Ω . The memristor has an initial value of 43 k Ω . By taking the on-resistance of the two switches into account ($R_{on} \approx 442 \Omega$ each), R_{C2} is calculated and equals to 5.96 k Ω (The R_{C1} and R_{C2} resistance are the best case choices according to simulation results).

The rest of the MSB resistors except R_6 are assigned with binary weighted resistance. For example, R_2 has a resistance of 21 k Ω , which is two times of the equivalent R_{CAL} . R_6 is designed to be the same as R_5 to work as a redundant bit. When combining the R-2R DAC with the binary-weighted DAC, R_7 has to be two times of R_6 to extend the binary relationship. An OTA is designed to establish a virtual ground to the binary-weighted DAC and the R-2R DAC, this ensures that resistors in the two DACs work separately and not causing any interaction. The relationships of the resistors are represented in the following equations.

$$R_2 = 2 \times (R_{C1} + R_{C2} // R_S) = 21 \text{ k}\Omega \quad (3.3)$$

$$R_i = 0.5 R_{i+1}, \text{ for } i = 2, 3, 4, 5 \quad (3.4)$$

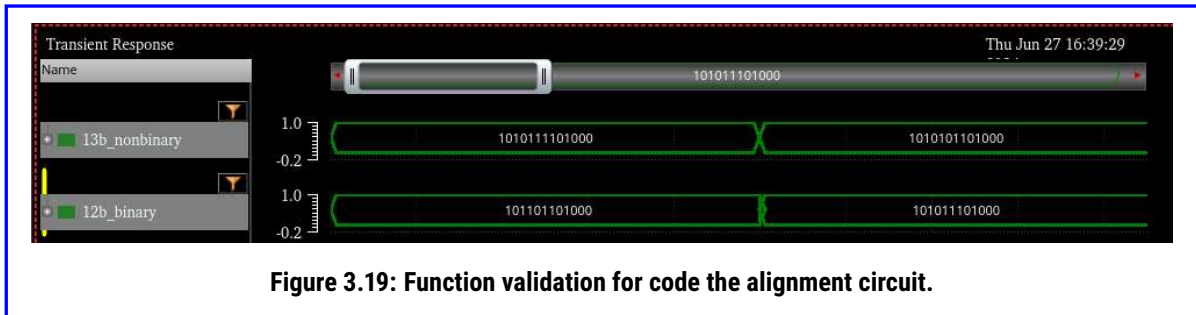
$$R_5 = R_6 = 168 \text{ k}\Omega \quad (3.5)$$

$$R_j = R_{j+1} = 2 \times R_6 = 336 \text{ k}\Omega, \text{ for } j = 7 - 13 \quad (3.6)$$

The total resistance of our proposed hybrid DAC is 4.1895 M Ω , which saves more than 90% of area and power compared to a standard binary-weighted architecture. This total resistance value also means a lower thermal noise contribution to the DAC output than the standard binary-weighted architecture.

3.4.8 Code Alignment Circuit

Figure 3.20 shows the implementation of the code alignment circuit, which can convert the 13-bit non-binary code to the binary 12-bit code. In this design, the 'MSB-5' (B_7) is used as the redundancy bit, to recover the 13-bit code to the 12-bit one, the code alignment circuit simply adds the B_7 to its preceding B_8 to get the 12-bit code. This arithmetic operation can be easily realised in the digital domain, which can be either designed by using coding (i.e., Verilog coding and then synthesizing.) or routing-by-hand methods. The overflow function is almost never be used in actual application because all the ADCs are seldomly used in full-scale mode. Figure 3.19 illustrates the simulation result which validates the function of the code alignment circuit. For the first 13-bit code input ('1010111101000'), the redundancy bit is '1', the resulting 12-bit code is '101101101000'. For the second 13-bit code input ('1010101101000'), the redundancy bit is '0', the resulting 12-bit code is '101011101000'. Both results prove that the designed code alignment circuit is able to achieve the desired function.



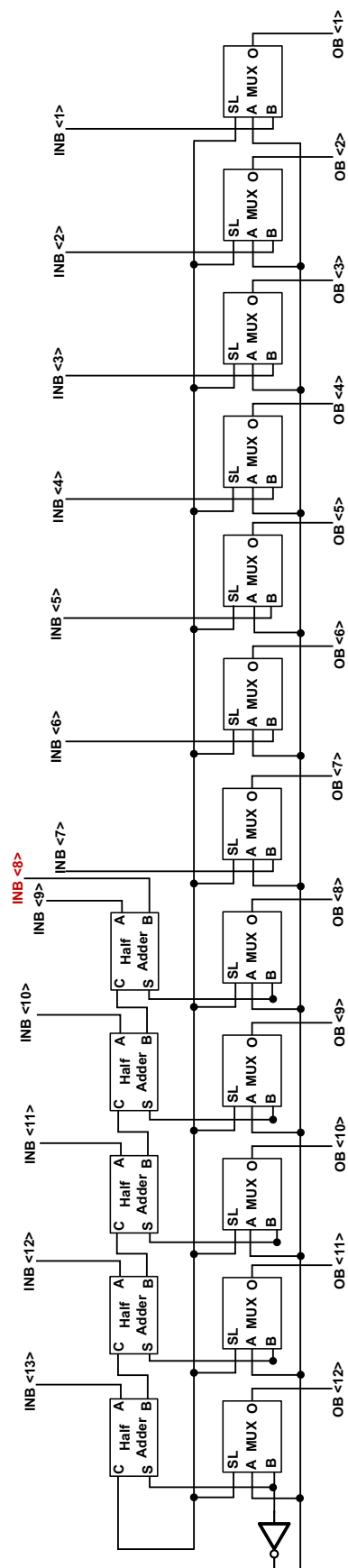


Figure 3.20: Implementation of the code alignment circuit.

3.5 Simulation Results of the proposed SAR ADC

3.5.1 Calibration Logic

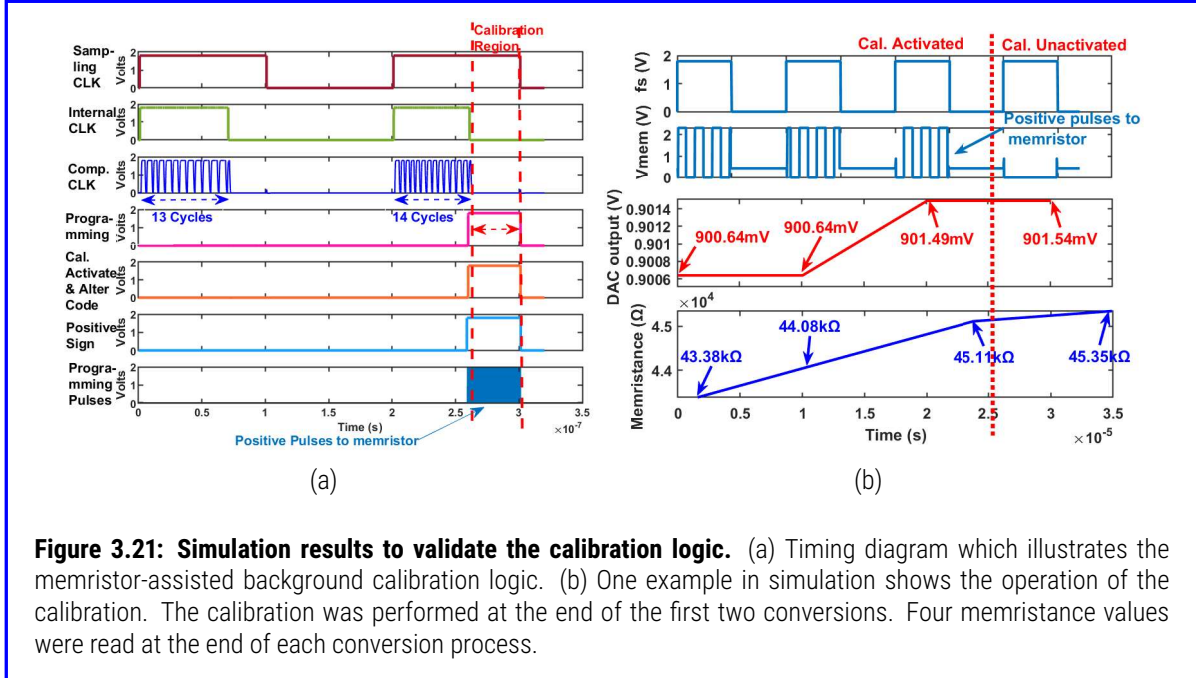
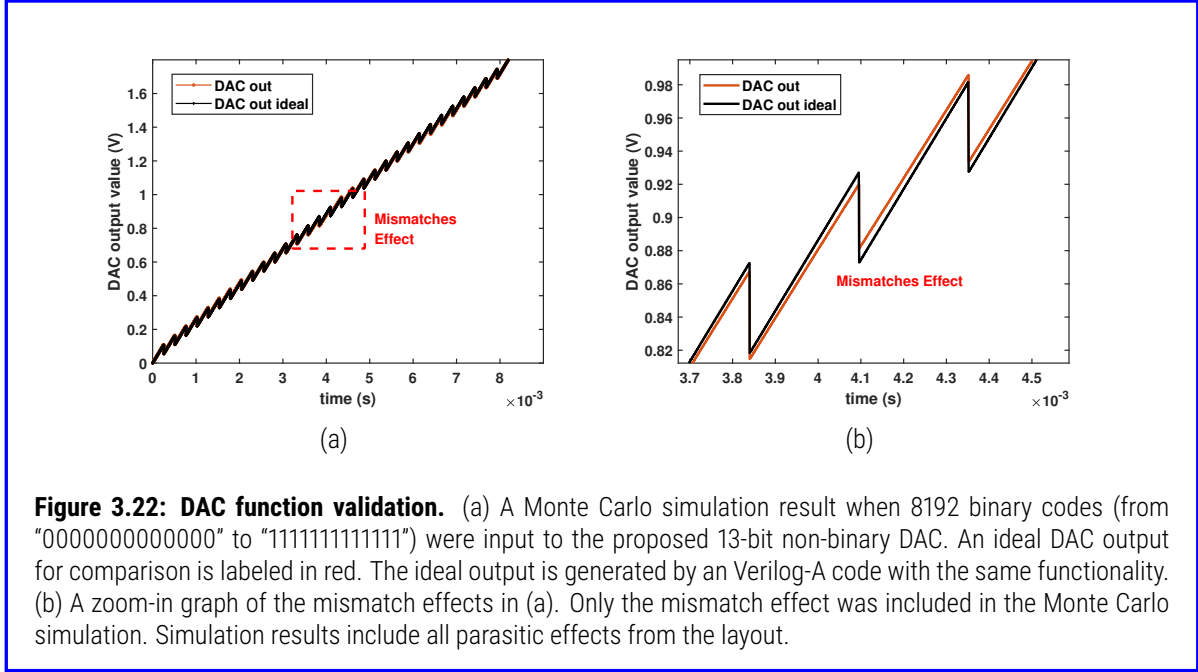


Figure 3.21: Simulation results to validate the calibration logic. (a) Timing diagram which illustrates the memristor-assisted background calibration logic. (b) One example in simulation shows the operation of the calibration. The calibration was performed at the end of the first two conversions. Four memristance values were read at the end of each conversion process.

Figure 3.21 shows an example of the simulated memristor-assisted background calibration timing diagram with the gradual change of the memristance and the DAC output. Figure 3.21 (a) demonstrates two scenarios. In the first conversion process, there is no error detected by the detection circuitry, so 13 normal conversion cycles are performed. The asynchronous SAR logic outputs the result when the last comparison is finished. In the second conversion process, a “calibration activation code” is detected and the calibration cycle is activated. 14 cycles (13 normal conversion cycles + 1 calibration cycle) are performed. After the 14th cycle, a positive sign is determined based on the sign error (Δ). The pulse generator then generates pulses with the corresponding polarity to program the memristor in the rest of the conversion period. Figure 3.21 (b) shows one example of how the gradual change of memristance trims the DAC output. V_{mem} is the voltage across the memristor, and DAC output is the output value of the hybrid binary/R-2R DAC after each A/D conversion. The sampling frequency is 100 kS/s and the input of the ADC is set to a fixed DC voltage of 901.7 mV (code: 100000xxxxxxx or 011111xxxxxxx) to ensure the MSB calibration is activated in all four conversions. The exact pulse width and amplitude are 800 ns and 2.2 V respectively (2.2 V pulses can present the result more obviously than 2V pulses). As shown in the result, in the first three conversions, the DAC output is lower than the input value, and the “calibration activation code” is detected in both conversions. Based on the sign, positive programming pulses are generated to adjust the memristance by a specific resistance value. Then, in the next conversion, the DAC output reaches 901.54 mV, which is closer to the desired level (less than 1 LSB), and this indicates that the calibration performed in the previous three conversions effectively calibrate the DAC. The

calibrated DAC assists the ADC to achieve more accurate results in the subsequent conversions, and thus improves the ADC output linearity. The amplitude and period of the programming pulses can be adjusted to achieve different calibration effects by trading off between calibration speed and power consumption (e.g., a higher amplitude pulse can speed up the calibration process but increasing the power consumption and reducing the calibration accuracy.).

3.5.2 Functionality of the DAC



To ensure the proposed DAC works as expected, its functionality was examined separately by performing a full-scale sweep with device models for Monte Carlo simulation (i.e., from "0000000000000" to "1111111111111"). As shown in Figure 3.22 (a), 8192 binary codes were input to two 13-bit non-binary DACs. One is the proposed hybrid resistive DAC, and the other one is an ideal DAC written in Verilog-A based on the functionality of the proposed DAC, which has no mismatch. Outputs of two DACs show continuous triangle-shaped steps, and these steps are expected because binary codes were used to test non-binary DACs. When the input approaches an equivalent analog value of 0.9 V, caused by the mismatches, the output of the proposed DAC starts to deviate from the ideal output steps. Figure 3.22 (b) shows a zoom-in graph of Figure 3.22 (a), it is obvious that the intrinsic element mismatches of the DAC lead to the output non-linearity, which deteriorates the ADC performance. Despite the errors caused by the mismatches, the proposed DAC performs the correct D/A conversion, and the non-linearity caused by mismatches can be calibrated by the proposed calibration scheme. Based on the proposed DAC architecture, when the memristance varies from $28 \text{ k}\Omega \pm 1 \text{ k}\Omega$ to $74 \text{ k}\Omega \pm 1 \text{ k}\Omega$, the DAC output can be adjusted by a range from - 7.33 LSB to + 10.08 LSB. (N.B., the LSB here is the LSB for the 13-bit non-binary DAC, which is calculated by $\frac{V_{ref}}{D_{in}}$.)

3.5.3 Improvements in Static Linearity

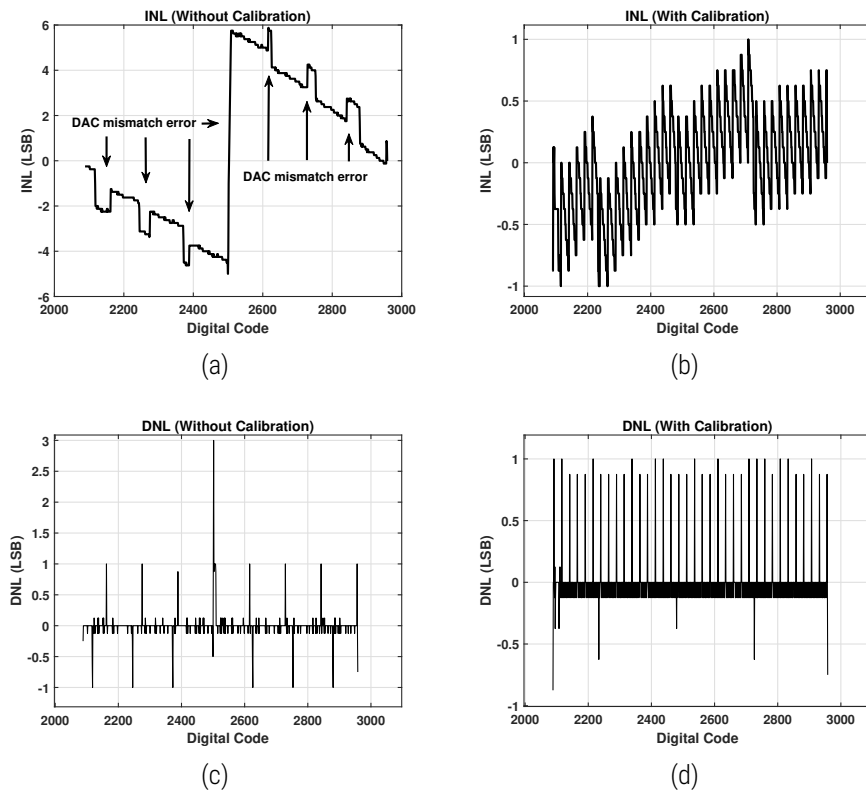
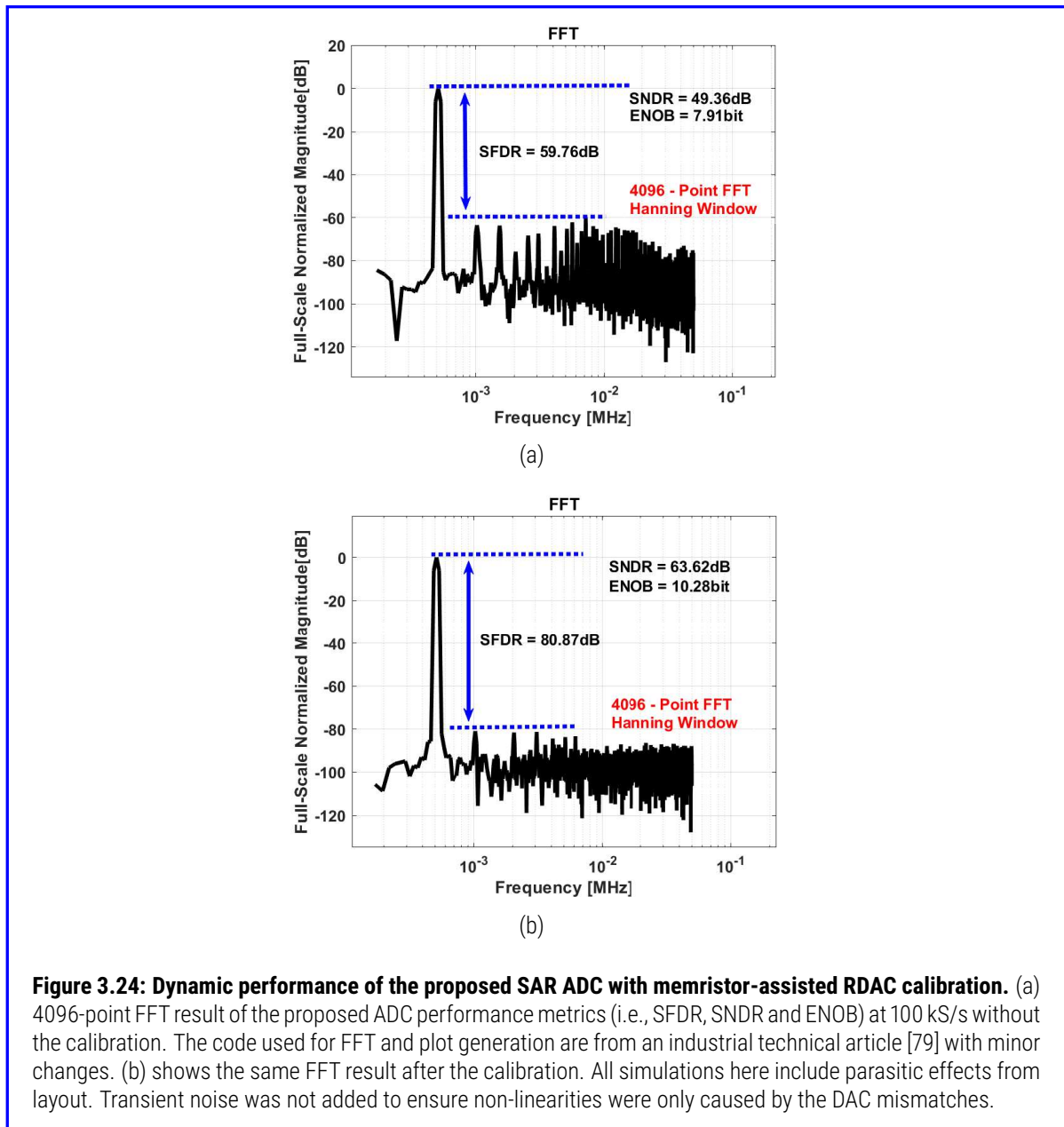


Figure 3.23: Improvements in Static linearity. (a) Simulated INL without the calibration. (b) Simulated INL with the calibration. (c) Simulated DNL without the calibration. (d) Simulated DNL with the calibration. The sampling rate is 100 kS/s. Transient noise was not added to ensure the non-linearities were only caused by the DAC mismatches. The amplitude and pulse width of programming pulses were set to 2 V and 500 ns respectively in the simulation. Simulation results include all parasitic effects from the layout. Test codes are shown in Appendix 5.1.

To demonstrate the efficacy of the proposed calibration scheme, histogram code density test [78] method were used to measure the INL and DNL before and after the calibration. The transient noise was disabled during the simulation and only the mismatch effect was included in the Monte Carlo simulation. This ensures the non-linearity of the ADC output can only be caused by intrinsic element mismatches. Please note that the statistical model of memristors is not yet available and therefore the Monte Carlo simulation presents the statistical variation of all the CMOS components in the circuits but not the memristors. However, the absolute precision of the memristors is not required in the proposed calibration scheme since their resistance values are adaptively programmed to null the error of DAC in a closed loop. As shown in Figure 3.23, all simulations were run at 100 kS/s while a ramp signal with a step size of 0.125 LSB is applied at the input, which corresponds to 8 conversions at each step (i.e., 8 times code appearance in each code bin.). The input range was set from 0.9 to 1.3 V because this is the range where the MSB calibration is activated. In this range, the 13-bit non-binary code "100000xxxxxxx" or "011111xxxxxxx" was detected and the effect of mismatches occurs alternatively because different resistors were activated for two different codes. A number of 7280 input codes, which indicate 910 (7280/8) 12-bit binary codes, were tested

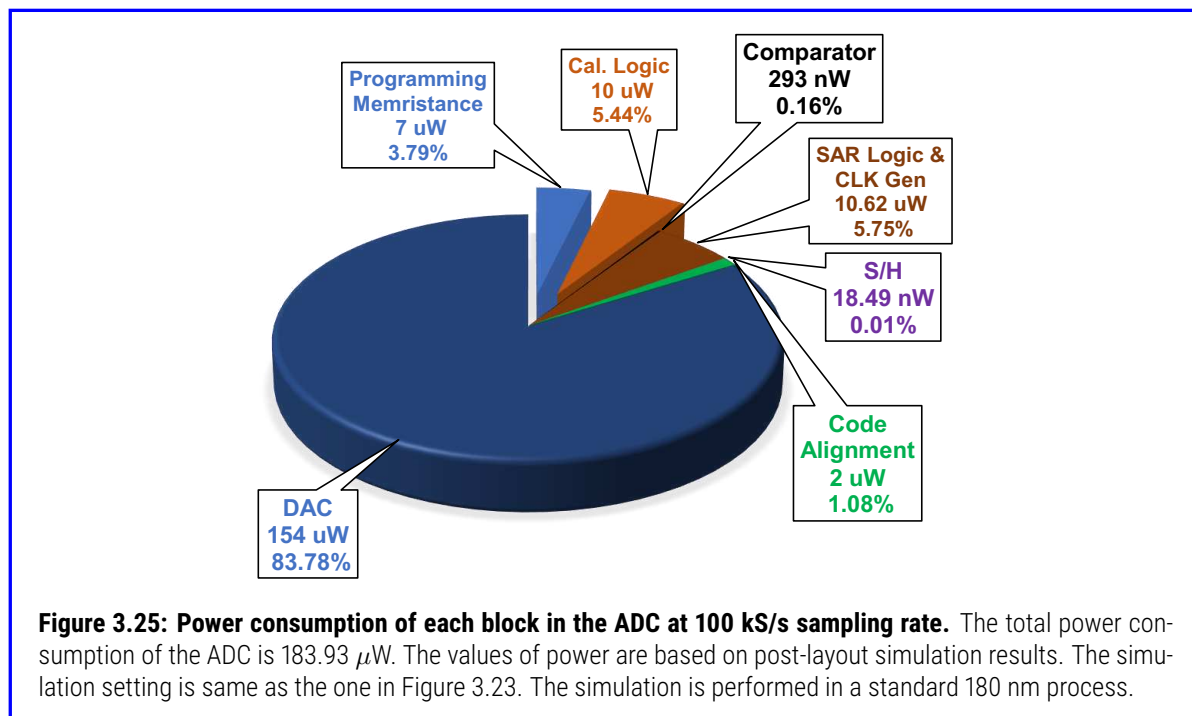
to verify the proposed scheme. The first 40 binary codes were omitted in the plots because the calibration was conducted during the first 320 conversions (40 binary codes $\times$ 8 = 320). From Figure 3.23 (a) and (b), it is apparent that the INL is effectively suppressed when the error caused by DAC mismatches is calibrated. Furthermore, as shown in Figure 3.23 (c) and (d), the DNL hit within ± 1 LSB with the aid of calibration. This indicates there are fewer missing codes after the calibration is performed. These results prove that the matching in DAC can be effectively improved with the proposed memristor-assisted calibration scheme.

3.5.4 Dynamic Performance and Walden Figure-of-Merit (FOM)



The FFT results were got by applying a nearly full-scale sine wave to the ADC, whose frequency was calculated by the coherent sampling (i.e., $\frac{f_{IN}}{f_{SAMPLE}} = \frac{N_{Window}}{N_{Record}}$) [77]. The sampling frequency f_{SAMPLE} is 100 kHz in this research, the number of the data points (N_{Record}) is 4096, and the integer number of cycles of the sine wave within the sampling window (N_{Window}) is 21. Thus, the input frequency of input sine wave signal is 512.695 Hz on the basis of the equation in [77]. After performing the FFT to the output of the ADC, a resulting FFT graph is presented in Figure 3.24 (b) (c). Figure 3.24 (a) illustrates the FFT result when the proposed calibration scheme is not active, while the result shown in (c) shows the FFT result after a period of calibration performed. According to the information provided in the FFT graph, the proposed ADC with the memristor-assisted background calibration has improved the SFDR from 59.76 dB to 80.87 dB, and the harmonic distortion level has been significantly reduced. The SNDR has been improved by 14.26 dB, which means a 2.37 ENOB improvement. The overall performance means a 1.459 pJ per conversion step in Walden FOM ($\frac{POWER}{f_s \cdot 2^{ENOB}}$) (For the power consumption measurement method, please check the next section). Although this result is not able to compete with the state-of-the-art, the proposed memristor-assisted background calibration scheme is able to improve the linearity performance of the single-ended SAR ADC, which demonstrating the efficacy of the proposed calibration scheme at the post-layout simulation level (Note: The FFT result shown here demonstrates the best case ADC performance compared to results got with other test signals.).

3.5.5 Power Consumption



A total of 32768 (4096×8) codes were swept. During the simulation, the “calibration activation codes”, which can be either “100000xxxxxxx” or “011111xxxxxxx”, were detected. Therefore, 2048 (1024×2) calibrations were activated, which means the calibrations were turned on in 6.25% ($2048/32768$) of the total simulation time. Because the calibration was not always on during the complete sweep, the power consumption of the calibration (Cal. Logic + Programming memristance) in 1 conversion is averaged over the full range and kept at a low value. It is worth noting that the memristor model in simulation cannot fully characterize the non-linear power consumption over time, and therefore an estimation has been made that the energy consumed during each voltage pulse when trimming the memristor based on the measurement data from the real memristors (the switching behaviour and I-V characteristics of the chosen memristor were measured in the group’s lab and reported in [80]). For this design, the amplitude of the programming pulses was set to 2 V, due to the change of memristance being within a certain range, the instant worst-case current through the memristor is accounted for and equals to 74 μ A, and thus the corresponding worst-case energy consumption within each trimming pulse is 148 pJ.

3.6 Chip Layout

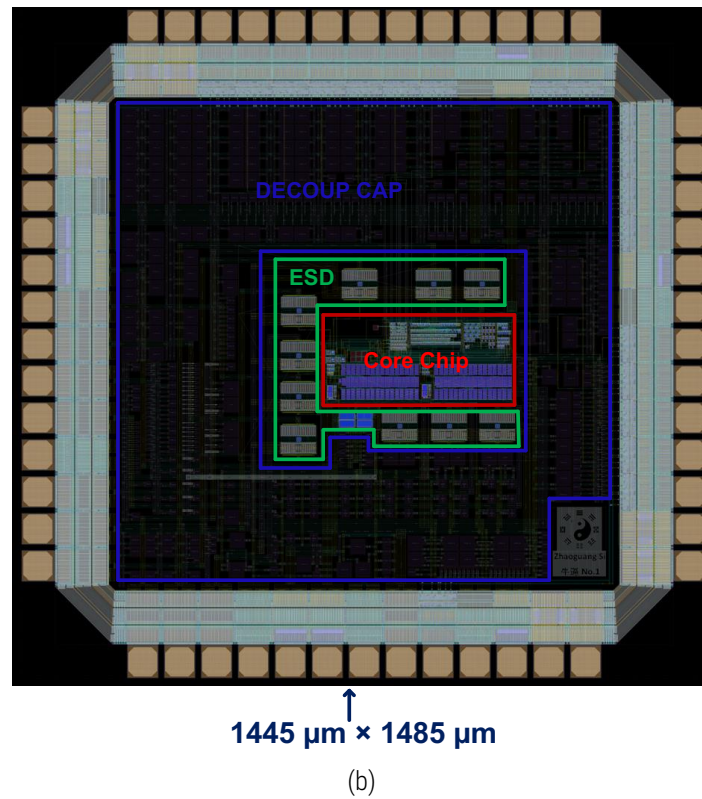
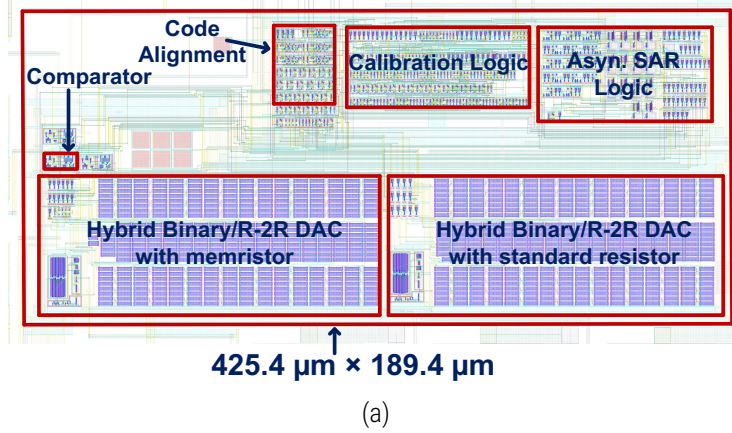


Figure 3.26: (a) Layout of the core chip. (b) Layout of the chip for the tape out.

Figure 3.26 (a) shows the layout of the core chip. The dimension of the core chip is 425.4 μm by 189.4 μm , and the total area occupation is 0.08 mm^2 . The memristor will be laid on top of the MSB resistors in the DAC via Metal 5 and Metal 6. Therefore, the memristor itself does not occupy any area. It is worth mentioning that the DAC area is larger than it should have been. This is because the tape-out chip was designed with an off-chip option to connect the memristor since this is the first version of the test chip, so it would be a much safer way to do so. Compared to the SAR ADC with sign-based calibration in [59], which adopted an area occupation of 0.0675 mm^2 in 40 nm

technology node, this design has larger area occupation which is mostly due to the fact that the design is implemented in a much older 180 nm technology. The final submitted layout for the tape out is shown in Figure 3.26 (b), which occupies an area of 1445 μm by 1485 μm .

3.7 Testbench Design & Setup

The fabricated chip microphotograph is shown in Figure 3.27. This section details the measurement setup and data acquisition system for characterizing the proposed ADC. The test infrastructure employs a Xilinx ZedBoard featuring a Zynq-7000 SoC (XC7Z020CLG484-1), which captures and temporarily stores the ADC output data in its internal processor before transferring it to a PC via UART communication. Subsequent data analysis is performed using MATLAB in PC platform.

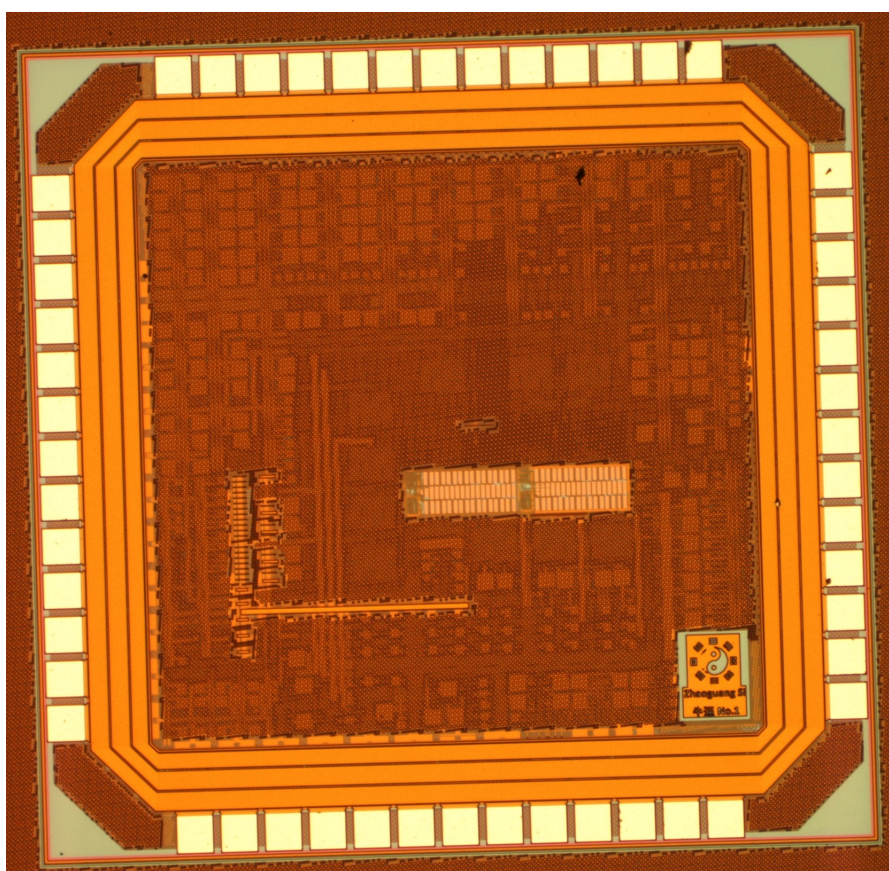
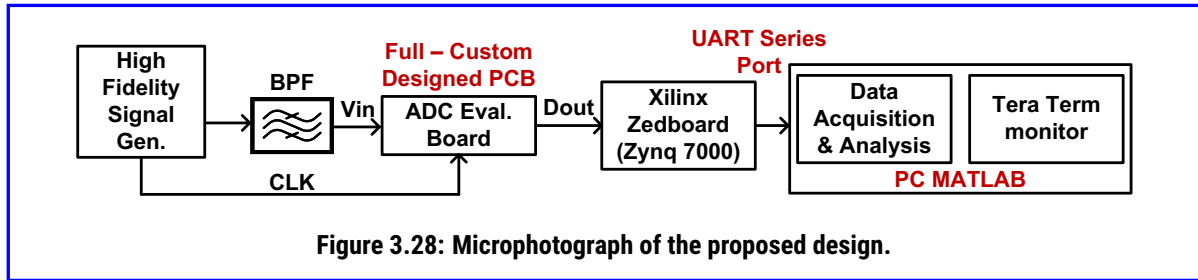


Figure 3.27: Microphotograph of the proposed design.

3.7.1 Top-level System of the testbench

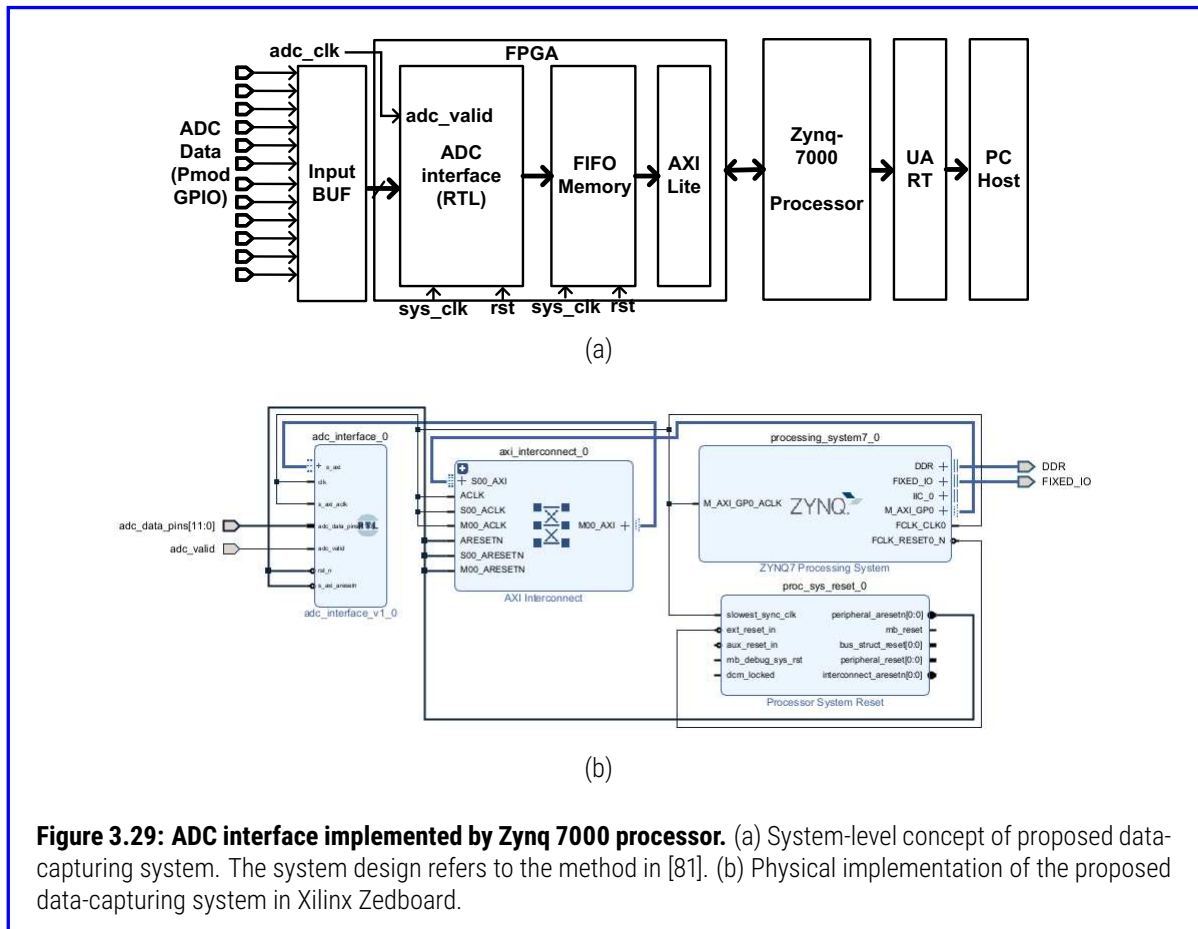
Figure 3.28 illustrates the ADC test architecture. The measurement setup requires a precision signal generator and optional bandpass filters to suppress out-of-band harmonics, ensuring the input signal's ENOB substantially exceeds the ADC resolution. The clock source must provide signals with minimal phase noise and jitter characteristics. A custom-designed evaluation PCB interfaces with the fabricated silicon, providing the necessary biasing and control signals for proper operation. The ADC's 12-bit digital output is acquired through GPIO ports and processed by a full-customized ADC interface module implemented in the Xilinx ZedBoard FPGA, with the captured data stored

in on-chip memory. Subsequently, the stored data is transferred to a PC via UART communication for analysis. The following subsections detail the custom ADC interface implementation and evaluation board design.



3.7.2 ADC interface design in Xilinx Zynq-7000

Due to the high cost of commercial ADC test equipment, this work uses a Xilinx Zynq-7000 SoC for data acquisition. Figure 3.29(a) illustrates the data capture system architecture. The ADC output interfaces with the ZedBoard through its Pmod GPIO headers (JA and JB), each providing 2x6 pin connectivity. A custom *adc_interface* module processes the incoming data and buffers them in the onboard FIFO memory. Subsequently, the Zynq-7000 processor transfers the captured data to a PC via UART communication.



The system implementation of the system in Figure 3.29(a) was performed on the Vivado platform, the *adc_interface* module is designed as an RTL module with Verilog, and the module source code is shown in Appendix 5.4. To map Pmod pins to the adc interface module, a pin constraint file (see Appendix 5.5) was added to configure the pin position, referring to the Zedboard user guide [82]. In addition, the Zynq processor module also needs to be configured properly to enable the UART interface. Figure 3.29 (b) shows the block diagram of the implemented system, demonstrating the realization of the architecture proposed in Figure 3.29 (a).

3.7.3 ADC evaluation board design

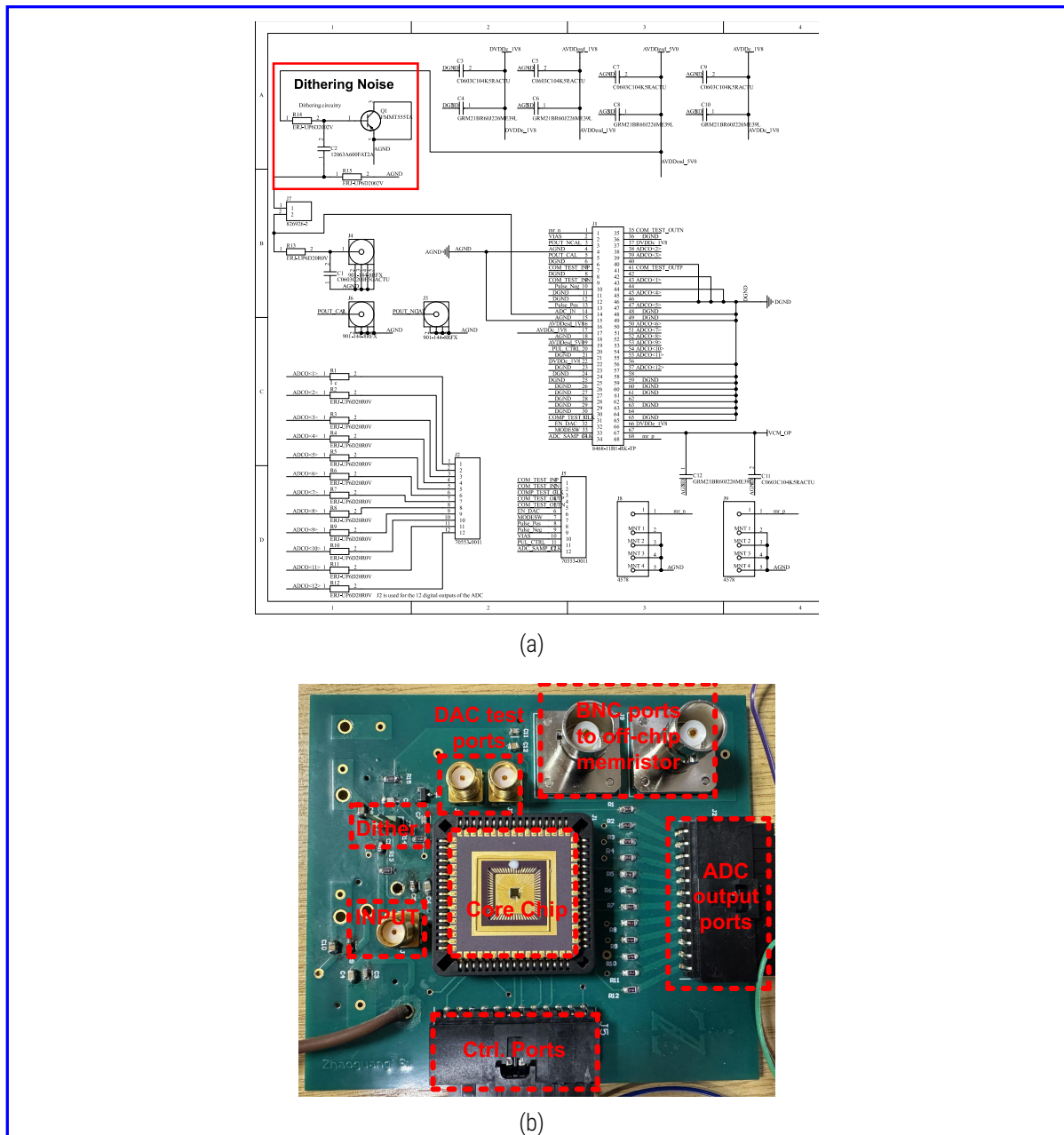


Figure 3.30: PCB design of the ADC evaluation board. (a) Schematic of the PCB board. (b) Printed PCB board.

The PCB design refers to the design method in [79], the PCB schematic is shown in Figure 3.30 (a), an optional dithering noise source was designed to improve the linearity performance of the overall design. Figure 3.30 (b) shows the printed PCB board with the fabricated core chip.

3.7.4 Testbench Integration

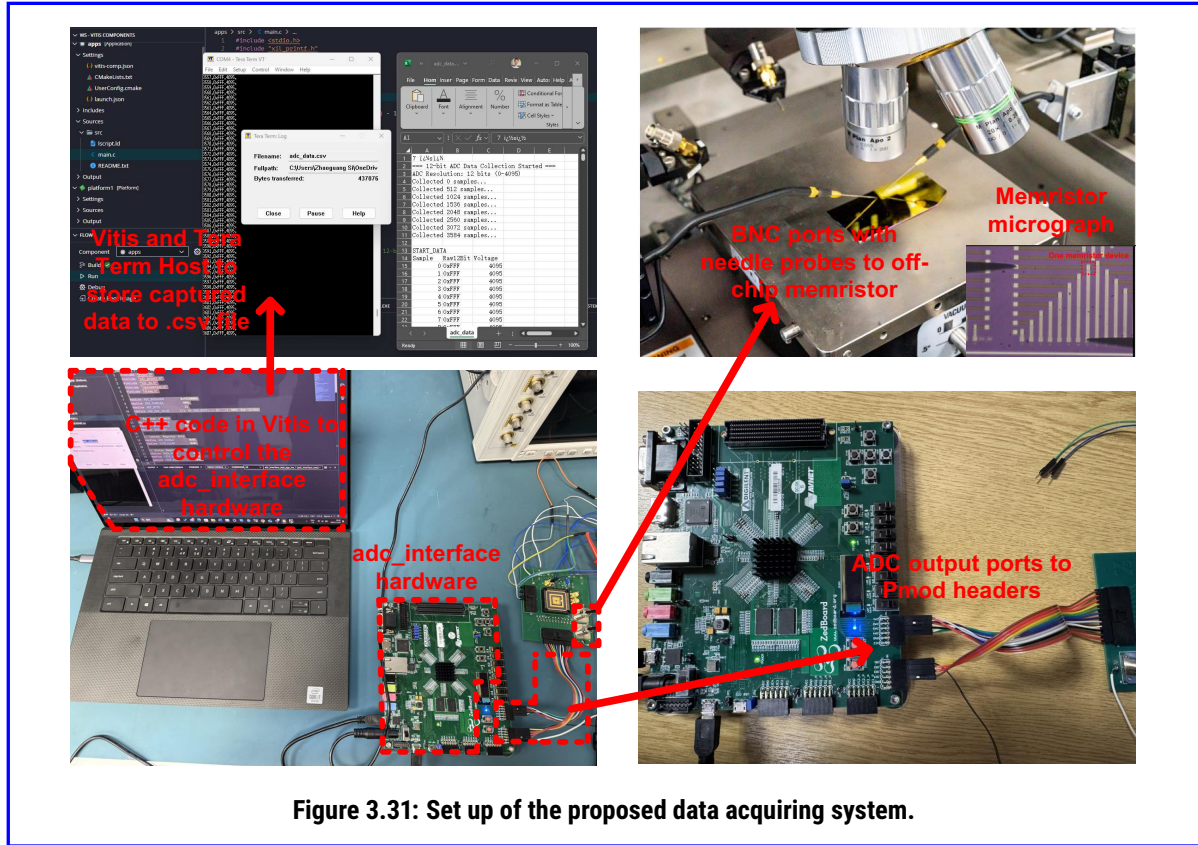


Figure 3.31: Set up of the proposed data acquiring system.

The data acquisition system, comprising custom-designed ADC interface hardware and PCB board, is illustrated in Figure 3.31. To ensure seamless integration and operation of all system components, a comprehensive interface configuration was implemented. The initial phase involved configuring the Vitis software environment to establish control over the ADC interface hardware. The corresponding control code is detailed in Appendix 5.6. Data capture and storage were facilitated through a terminal emulator (Tera Term), configured with a baud rate of 115200 bps, which logged the acquired data to CSV format. Then, the ADC outputs were interfaced with the Zedboard through its Pmod headers using appropriate wiring connections. As illustrated in Figure 3.31, these connections were established according to the pin configuration specifications defined in the XDC file (detailed in Appendix 5.5). Successful data acquisition from the ADC outputs through the Pmod headers is indicated by the illumination of the LD9 indicator on the Zedboard. Subsequently, the off-chip memristor was connected through needle probes on the probe station, and the needle probes with male BNC connectors were connected to female BNC connectors on PCB board. The pre-electroforming of the memristor can be done by commercial ArC ONE memristor characterization platform [40].

3.7.5 Measurement Challenges due to Memristor Device Limitations

Despite the robust design and data acquisition capabilities of the chip measurement system, significant challenges emerged due to memristor device availability constraints. The foremost challenge originated from the extended fabrication cycles required for on-chip memristors, which exceeded feasible research timelines. The measurement methodology was subsequently modified to incorporate off-chip standalone memristors for calibration procedures, acknowledging the associated performance trade-offs. Furthermore, the low yield rates of standalone memristors presented an additional obstacle, as the available laboratory devices exhibited substantial deviations from VerilogA model predictions. These implementation challenges impeded the acquisition of empirical performance data comparable to post-layout simulation results. Future measurements with enhanced validity are anticipated as memristor fabrication technology matures and manufacturing efficiency improves.

3.8 Discussion and Conclusion

3.8.1 Discussion

This research has demonstrated the feasibility of using the memristor to calibrate the intrinsic element mismatches in the RDAC of the SAR ADC. The results shown in Section 3.5 prove that the proposed scheme is a viable solution to improve the output linearity of a high-resolution SAR ADC. Since this is a feasibility study, the research mainly focused on functionality verification and there are still limitations that need to be overcome before a memristor-assisted SAR ADC can be developed which outperforms the state of the art.

First of all, although the CMOS/Memristor integration technology is rapidly progressing, available memristor models today do not yet include process variation, mismatches, and stochastic switching behaviour. Accurately programming the memristance with voltage pulses is also critical because the memristors may experience device-to-device, cycle-to-cycle variations, and retention degeneration. The proposed scheme works based on a closed-loop mechanism which tackles to some extent the effects of these variations, but a more exhaustive study is only possible as more data becomes possible after these behaviours of memristors are characterized on a large scale (e.g. using the characterization platform developed in [83]). Another limitation of the memristor is that it needs a certain amount of time to be programmed; the proposed design can not be adopted to high-speed SAR ADCs at the moment (100 kS/s for this design, despite of a 10 MS/s without the memristor-assisted calibration turned on). Yet, as mentioned in Section 3.3, using a memristor in a coarse or fine trimming affects the speed and power consumption, the speed of the future SAR ADCs employing the memristor-assisted calibration can be adjusted according to different applications (e.g., the neural interface requires a fine trimming thus lower speed).

In this study, the resistive DAC was used so that the element trimming using memristors is more straightforward. However, resistive DACs are typically not as energy-efficient as capacitive DACs (e.g., in this design, the proposed RDAC has a power consumption of 154 μ W, which accounts for 83.78% of the total ADC power), and the single-ended architecture is very sensitive to comparator kickback noise, comparator input parasitic capacitance and the noise from the power supply, deteriorating both static and dynamic linearity performance of the ADC. Using memristors to calibrate capacitive DACs can be an interesting study topic in the next step, and a possible method is to use memristors to alter the DAC settling behaviour to effectively cancel the errors. The limitations of the RDAC have become the motivation of the research in the next chapter, which is exploiting the memristor to do the CDAC calibration (Please see more details in the next chapter's introduction.). Besides, as mentioned in Section 3.4, the future design also plan to use a memristor array in the resistive DAC to trim more MSBs (i.e., MSB-1, MSB-2, etc.) instead of just the MSB. In such way, more errors caused by mismatch will be calibrated, and peripheral circuits can be reused to trim each memristor to reduce the circuit area and power consumption.

In the end, although the proposed SAR ADC still has limitations and is not yet competitive compared with the state-of-the-art ADCs, the author believes the work is valuable to the memristor research community since this is the first work using memristor as a trimming element to calibrate a high-resolution SAR ADCs. With the results reported the author would like to draw attentions of the ADC design community to an alternative calibration scheme enabled by the emerging memristor technology. The author expects the performance will improve in the near future as the memristor technology is advancing rapidly. Although these defects bring concerns to the memristor-assisted circuit designs, the proposed architecture is still a suitable candidate among the existing calibration techniques as the technology will reach a mature standard eventually. The technical challenges faced by the author today may be resolved in the near future with the global development of CIM devices (e.g., a memristor device with lower programming voltages and faster programming speed, which can reduce the time for correction process in the calibration loop).

3.8.2 Conclusion

This work presents a novel calibration scheme that exploits an in-house fabricated memristor to calibrate the intrinsic DAC mismatch of a 12-bit SAR ADC. To the best of the author's knowledge, this is the first reported memristor-assisted calibration scheme for ADCs. The proposed scheme is designed based on the sign-based background calibration method along with the memristor that is used as both the calibration information register and the trimming element which is facilitated by using a hybrid binary/R-2R DAC architecture. The proposed scheme is validated to be effective to improve the linearity performance of Nyquist SAR ADC. Table 3.2 shows the comparison table among different SAR ADCs with sign-based calibration scheme, it can be seen the proposed scheme's efficacy in this work is relatively competitive in 180 nm process, despite the speed and power consumption are limited by the RDAC based SAR ADC architecture. While this is a feasibility study, the proposed scheme is expected to improve the SAR ADC performances further as memristor technology advances further.

Table 3.2: Performance comparison

	JSSC'17 [59]	This work [2]	ISCAS'25 [84]
Architecture	SAR ADC with sign-based calibration		
Technology (nm)	40	180	65
Area (mm^2)	0.0675	0.0531*	0.0153
ENOB (bit)	10.40	10.28**	10.67
Supply voltage (V)	1.0	1.8	1.0
Sample rate (MS/s)	6.4	0.1	5.0
SNDR (dB)	64.10	63.62**	65.96
SFDR (dB)	81.90	80.87**	85.93

* Counting 1 extra DAC for testing.

** Results obtained from additional simulations.

Method in [59] is aligned to a US patent [85], and used in [86–88].

Memristor-assisted Background Calibration for a 12-bit SAR ADCs with CDAC

As discussed in Chapter 3, the single-ended SAR ADC with RDAC architecture is less efficient in terms of area and power compared to the existing state-of-the-art SAR ADCs with differential MOM CDAC architecture. Additionally, the single-ended design is highly susceptible to issues such as comparator kick-back noise, comparator input parasitic capacitance, and power supply noise, all of which adversely affect the ADC's static and dynamic linearity performance. Therefore, incorporating a memristor-assisted background calibration scheme into the SAR ADC with CDAC presents a promising research topic. In this chapter, a reference voltage (V_{ref}) compensated memristor-assisted background calibration is proposed. This calibration scheme is implemented in a 12-bit SAR ADC utilizing a V_{cm} -based CDAC switching scheme. The detection of capacitor mismatches is performed similarly to the method outlined in [59], allowing for the extraction of the mismatch error's sign. By exploiting the error sign, the mismatch can be calibrated through a feedback loop, which comprises two memristor-based voltage dividers. By adjusting the resistance of the memristor, the voltage divider is able to provide programmable V_{ref} voltage to the calibration bits of the CDAC. Consequently, the updated V_{ref} can trim the CDAC output level, mitigating the non-linearity caused by capacitor mismatches. Unlike the RDAC calibration, the tunable resistance of the memristor device does not directly affect the CDAC operation, thereby reducing the risk of memristor fabrication and improving the circuit reliability. The feasibility of the proposed scheme is validated by performing the post-layout simulations, the static linearity of the 12-bit SAR ADC has been effectively optimized, where the signal-to-noise-and-distortion ratio (SNDR) has been improved from 41.85 dB to 65.98 dB, and spurious-free dynamic range (SFDR) has been improved by 32.77 dB. The proposed SAR ADC occupies 0.0153 mm² area according to the layout floorplan, making it a promising candidate for miniature/large-scale sensor interface application.

The work proposed in this chapter is an extended work of the author and his supervised master student Sabrina Wang's previous work done in a standard 180 nm CMOS process. All the extended work proposed in this chapter is an improved version of the previous work and re-designed by the author in a standard 65 nm CMOS process. This work has been accepted by **2025 IEEE International Symposium on Circuits and Systems (ISCAS) [84]**. © 2025 IEEE.

4.1 Introduction

In Chapter 3, the memristor-assisted calibration scheme was realised in a resistive digital-to-analog converter (RDAC) since the DAC output level can be straightforwardly trimmed by the memristor. However, the single-ended SAR ADC with RDAC architecture presents several challenges that render it less suitable for compact and low-power array-based interface applications, such as radars, image sensors, [89] and neural interfaces.

The first issue is that designing a differential architecture for the RDAC-based SAR ADC incurs significant overhead in power and area (one extra RDAC ladder and holding capacitor), which is undesirable for compact and low-power designs (differential RDAC-based SAR ADC is generally used for high-speed precision applications and designed in hybrid architecture [90]). The single-ended architecture directly refers the noise from " V_{CM} " source to the DAC output (see Figure 4.1 (a)), resulting in a poor Power-Supply-Rejection-Ratio (PSRR) and sensitive to electromagnetic interference (EMI). Secondly, the non-ideal effects, such as the kick-back noise and the clock feedthrough, from a high-speed dynamic-latch type comparator can feedback to its single-ended input, potentially causing incorrect comparison result. Thirdly, adjusting the gain of the single-ended DAC output to precisely match the input signal (V_{in}) range is difficult, which can lead to error codes. Additionally, directly integrating memristors into RDACs can lead to higher fabrication risks due to a low-yield process. These non-ideal effects limit the overall performance of the single-ended RDAC-based SAR ADC.

With the advancement of modern semiconductor technology nodes, CDAC-based differential SAR ADCs can achieve good linearity performance while maintaining low power, area, and latency costs [74, 89, 91]. As shown in Figure 4.1 (b), the differential architecture is able to cancel out the differential non-idealities from the comparator and source, improving the PSRR and linearity performance. The gain error caused by parasitic capacitance does not affect the comparison result as two CDAC ladders have the same parasitics, and two differential DAC outputs have the same gain errors, which can improve the accuracy of the comparison result. Besides, different types of full-customized state-of-the-art MOM capacitor [74, 89, 91] can achieve very small capacitance value (in the aF range) and area occupation, while maintaining a good matching among different unit capacitors, improving the overall DAC performance. To effectively benchmark the proposed memristor-assisted background calibration scheme against state-of-the-art SAR ADCs in array-based interface applications, it is promising to integrate the memristor-assisted background calibration scheme in a differential CDAC-based SAR ADC.

To further exploit the advantages of a memristor-assisted background calibration scheme, this chapter proposes a memristor-assisted CDAC background calibration technique. The memristor model used in this work is based on a stack-based memristor ($Pt/TiO_x/Al_xO_y/Pt$) as validated in [39]. A sign-based detection and voltage-compensated calibration method is employed to trim

capacitor mismatches. The proposed scheme is implemented in a 12-bit SAR ADC with a sampling rate of 5 MS/s, designed in the V_{cm} -switching scheme, and using a CDAC to demonstrate the feasibility of memristor-based CDAC calibration at the system level. Simulation results show that the linearity of the 12-bit ADC, including capacitor mismatches, is improved. Specifically, the SNDR and spurious-free dynamic range (SFDR) are enhanced by 24.13 dB and 32.77 dB, respectively. Additionally, the overall chip area is reduced by approximately 4.5x compared to [2, 59].

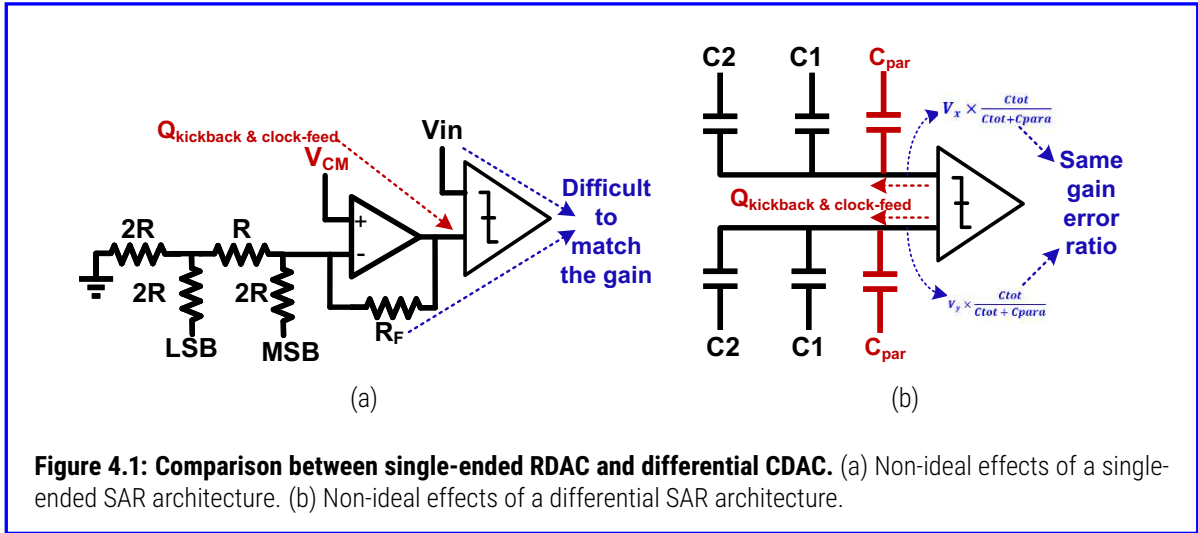


Figure 4.2 shows the energy consumption comparison among different SAR ADC switching schemes. The V_{cm} -based switching scheme was used in this design because it has the best power efficiency compared with other switching schemes. A detailed derivation of energy consumption among different switching schemes is shown as follows:

The energy of switching behaviour can be calculated by:

$$E_{tot} = P \cdot t = V \cdot I \cdot t \quad | \quad I \cdot t = C \cdot \Delta V \quad | \quad E_{tot} = C \cdot V \cdot \Delta V \quad (4.1)$$

For all switching schemes, only energy consumed in first two conversion cycles are demonstrated here for a brief view of calculation method.

Trial & Error switching scheme:

According to charge conservation, the voltage at the top plate of the DAC capacitor arrays in first two conversion cycles are (assuming the first comparator output is '1'):

$$\begin{aligned} V_X(1) &= 1/2V_{ref} + V_{CM} - V_{inp} \quad | \quad V_Y(1) = 1/2V_{ref} + V_{CM} - V_{inn} \quad | \\ V_X(2) &= 1/4V_{ref} + V_{CM} - V_{inp} \quad | \quad V_Y(2) = 3/4V_{ref} + V_{CM} - V_{inn} \end{aligned}$$

Then,

$$E_{0 \rightarrow 1} = -V_{ref} \times \{4C \times [(V_X(1) - V_{ref}) - (V_{CM} - V_{inp})] + 4C \times [(V_Y(1) - V_{ref}) - (V_{CM} - V_{inn})]\} \\ = 4CV_{ref}^2$$

$$E_{1 \rightarrow 2,1} = -V_{ref} \times \{4C \times [(V_X(2) - V_{ref}) - (V_X(1) - V_{ref})] + 2C \times [(V_X(1) - V_{ref}) - (V_X(1) - 0)] + \\ 2C \times [(V_Y(1) - V_{ref}) - (V_Y(1) - V_{ref})]\} = CV_{ref}^2$$

Energy-saving switching scheme is similar to the conventional one but with a halved Vref value, thus the energy consumption of the first two conversions are $2CV_{ref}^2$ and $0.5CV_{ref}^2$.

Monotonic switching scheme:

The first conversion of the monotonic switching scheme does not contain any energy consumption regarding the switching behaviour, thus the energy consumption is 0. According to charge conservation, the voltage at the top plate of the DAC arrays in the first two conversion cycles are (assuming the first comparator output is '1'):

$$V_X(1) = V_{ip} \parallel V_Y(1) = V_{inn} \parallel V_X(2) = V_{ip} - 1/2V_{ref} \parallel V_Y(2) = V_{inn}$$

Then in the second conversion cycle:

$$E_{1 \rightarrow 2,1} = V_{ref} \times \{2C \times [(V_X(2) - 0) - (V_X(1) - V_{ref})] + 4C \times [(V_Y(2) - V_{ref}) - (V_Y(1) - V_{ref})]\} = CV_{ref}^2$$

The **Vcm-based switching scheme** can be simply achieved by replacing the Vref in the monotonic switching scheme to Vcm, thus the energy consumption in the first two conversion cycles is $1/2CV_{ref}^2$, which is very energy efficient.

Based on the above calculation methods, the total energy consumption in all switching conditions can be easily calculated. By the assistance of the MATLAB, Figure 4.2 demonstrates the energy consumed by different switching schemes, it is apparent that the Vcm-based switching scheme is the most energy-efficient one.

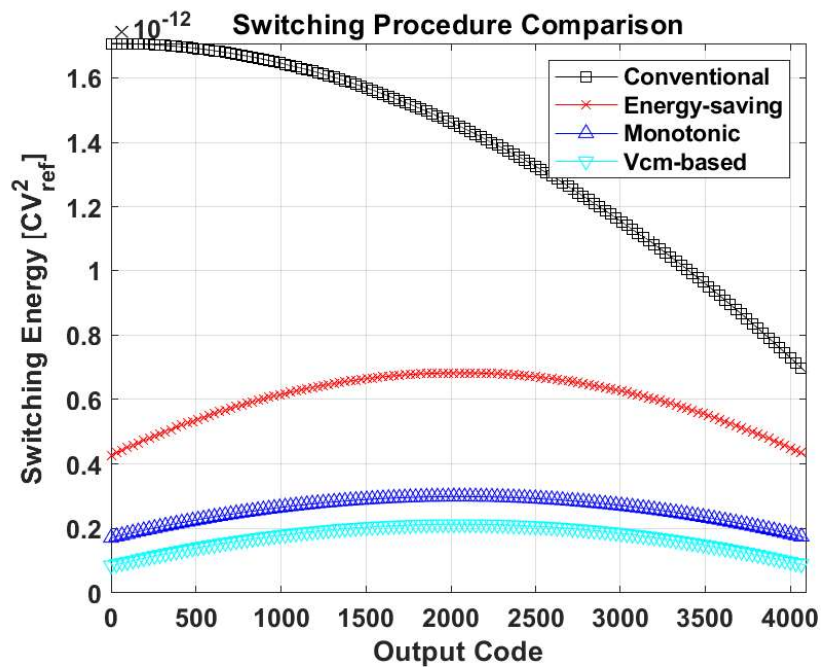


Figure 4.2: Comparison among total switching energy consumed by different 12-bit SAR ADCs with different switching schemes. The figure was generated by MATLAB scripts which represents the mathematical form of energy consumption of different switching schemes. The Vcm-based switching scheme consumed the least power compared to other existing schemes.

4.2 The Concept of the Proposed Methodology

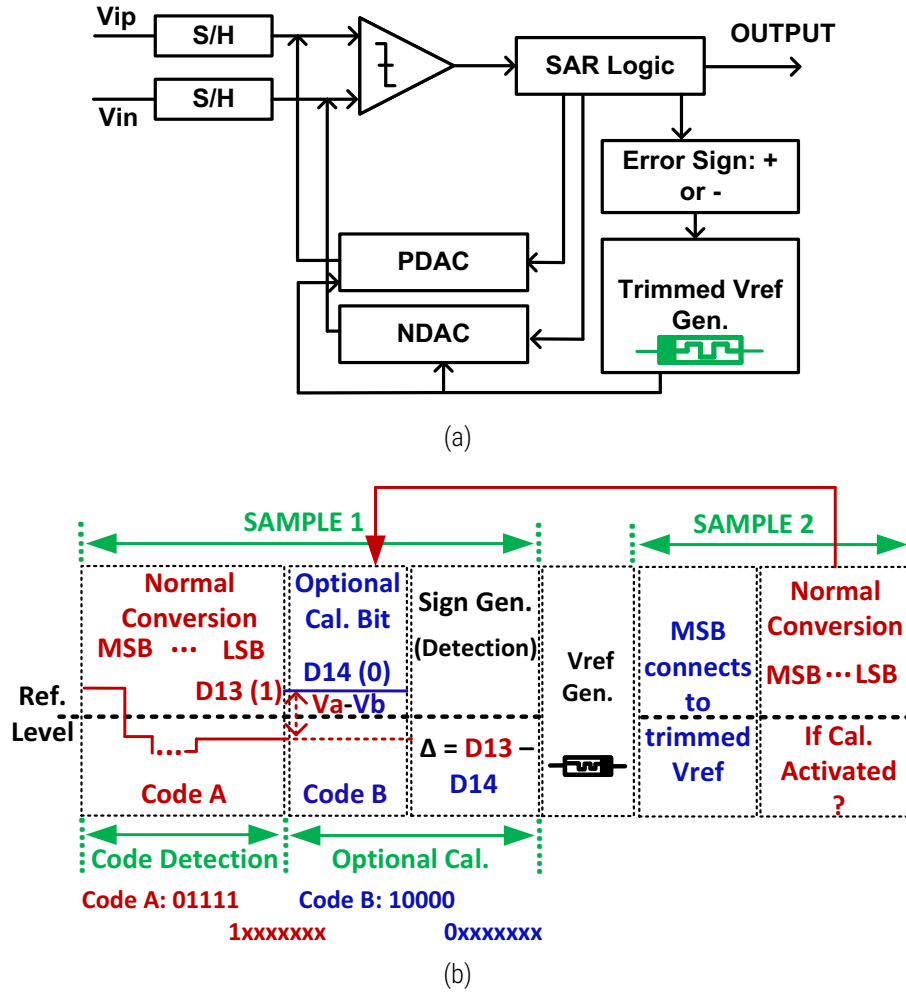


Figure 4.3: Memristor-assisted calibration scheme for CDAC-based SAR ADC. (a) The concept of the proposed memristor-assisted calibration scheme for CDAC-based SAR ADC. (b) An example of the logic flow of the memristor-assisted sign-based calibration scheme for CDAC. 13-bit non-binary Code A and Code B stand for the same 12-bit binary code: "10000xxxxxxx". Thus, ideally, the equivalent analog value of Code A and B equal to each other. The assumption here is only the MSB calibration is introduced.

Figure 4.3 shows the calibration concept of the memristor-assisted background calibration for CDAC-based SAR ADC. Assume only MSB calibration is considered here, as shown in Figure 4.3 (a), instead of using the memristor directly calibrate the DAC output level, the memristor is used to provide trimmed V_{ref} to trim the DAC output. Figure 4.3 (b) shows the logic flow of the proposed calibration scheme. The sign of the error is detected and generated in exact the same way as the one used in RDAC calibration. During the normal conversion, the ADC processes from MSB to LSB following standard SAR operation. The calibration scheme introduces an optional calibration bit (D14) after the normal 12-bit conversion cycle. This additional bit enables the detection of capacitor mismatches by comparing two specific digital codes: Code A (01111xxxxxxx) and Code B (10000xxxxxxx). These codes, ideally representin the same analog value, provide crucial

information about the capacitor mismatch when their actual analog values differ.

The calibration sequence consists of two main phases:

- **Code Detection:** The ADC first performs a normal MSB-to-LSB conversion (Code A). If calibration activation conditions are met, an additional conversion with the alternative code (Code B) is executed.
- **Error Sign Generation:** The difference between D13 and D14 ($\Delta = D13 - D14$) determines the direction of required calibration. This error sign guides the adjustment of V_{ref} through the memristor-based voltage generator.

Capacitor mismatch manifests as deviations in the ratio of each capacitor $C_{bit,n_{th}}$ to the total capacitance of the DAC, C_{total} , affecting the differential output voltages V_x and V_y . Rather than exploiting a parallel correction DAC alongside the main DAC, this work introduces a novel approach adding a compensating voltage to V_{ref} . The process is demonstrated using a 3-bit ADC example in Figure 4.4, where the charge conservation principle governs the sampling and redistribution phases.

During sampling, the total charge equations are:

$$\begin{cases} Q_{1P} = (V_{IP} - V_{CM}) * 4C = (V_x(1) - V_{CM}) * 4C \\ Q_{1N} = (V_{IN} - V_{CM}) * 4C = (V_y(1) - V_{CM}) * 4C \end{cases} \quad (4.2)$$

MSB (Assume the first comparison result is 0 ($V_x > V_y$)):

$$\begin{cases} Q_{2P} = V_x(2) * 2C + (V_x(2) - V_{CM}) * 2C \\ Q_{2N} = (V_y(2) - V_{ref}) * 2C + (V_y(2) - V_{CM}) * 2C \end{cases} \quad (4.3)$$

After charge redistribution and applying the principle of charge conservation ($Q_{1P} = Q_{2P}$, $Q_{1N} = Q_{2N}$, $V_{ref} = 2V_{CM}$), the resulting relationship are:

$$V_x = V_{IP} - \frac{V_{ref}}{4} \quad V_y = V_{IN} + \frac{V_{ref}}{4} \quad (4.4)$$

With the proposed calibration scheme, after the MSB comparison, the trimmed V_{ref} is applied to the bottom plate of MSB capacitor before the next comparison, then the charge redistribution becomes different from the normal one. For the P-side of the differential DAC, as there are no changes on the applied voltages, from $Q_{1P} = Q_{2P}$, $V_x = V_{IP} - \frac{V_{ref}}{4}$, which is exactly the same as the normal one. For the N-side:

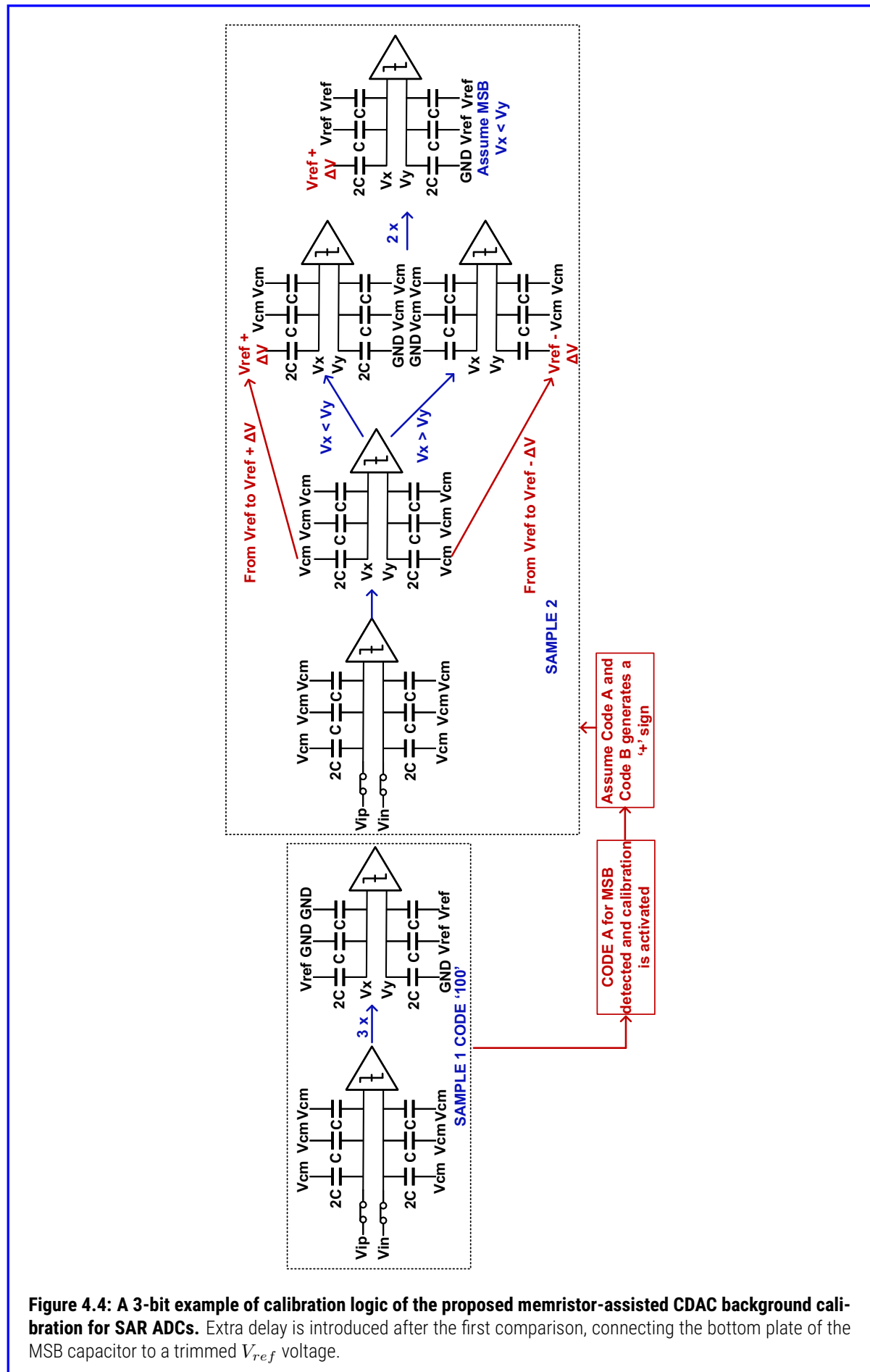


Figure 4.4: A 3-bit example of calibration logic of the proposed memristor-assisted CDAC background calibration for SAR ADCs. Extra delay is introduced after the first comparison, connecting the bottom plate of the MSB capacitor to a trimmed V_{ref} voltage.

$$Q_{2N} = (V_y(2) - (V_{ref} \pm \Delta V)) * 2C + (V_y(2) - V_{CM}) * 2C \quad (4.5)$$

Since $Q_{1N} = Q_{2N}$ still holds because there is no path for the sampled charges to flow. The V_y after the charge redistribution becomes:

$$V_y = V_{IN} + \frac{V_{ref}}{4} \pm \frac{\Delta V}{2} \quad (4.6)$$

The same approach can extend to V_x when $V_x < V_y$. This allows for the trimming of the DAC output level, effectively reducing the non-linear effects caused by capacitor mismatches. The V_{ref} can be dynamically adjusted during each sample conversion, meaning that a compensating voltage of $(\frac{\Delta V}{2})$ is applied to the DAC output.

The error detection mechanism employs four possible scenarios, as outlined in Table 4.1. After completing 13 conversion cycles, the detection of either Code A or B triggers an alternative code conversion. The resulting B13 and B14 values determine the calibration direction according to the look-up table, enabling precise adjustment of the memristor-based V_{ref} generator.

Table 4.1: Look-up table for MSB calibration direction

original code	B_{13}	alternative code	B_{14}	Sign Error
10000xxxxxxx	1	01111xxxxxxx	0	− ↓
10000xxxxxxx	0	01111xxxxxxx	1	+ ↑
01111xxxxxxx	1	10000xxxxxxx	0	+ ↑
01111xxxxxxx	0	10000xxxxxxx	1	− ↓

4.3 Implementation of the 12-bit SAR ADC with the Memristor-assisted Background Sign-based Calibration Scheme - CDAC

4.3.1 SAR ADC Top-level Architecture

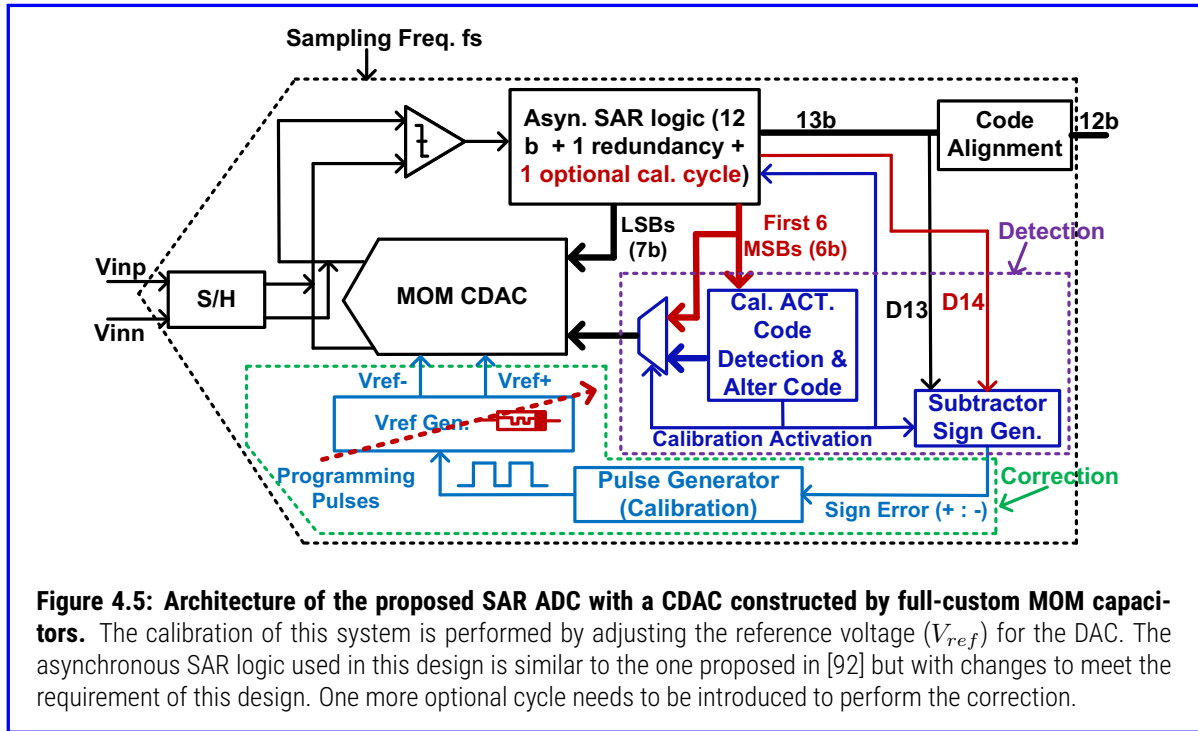
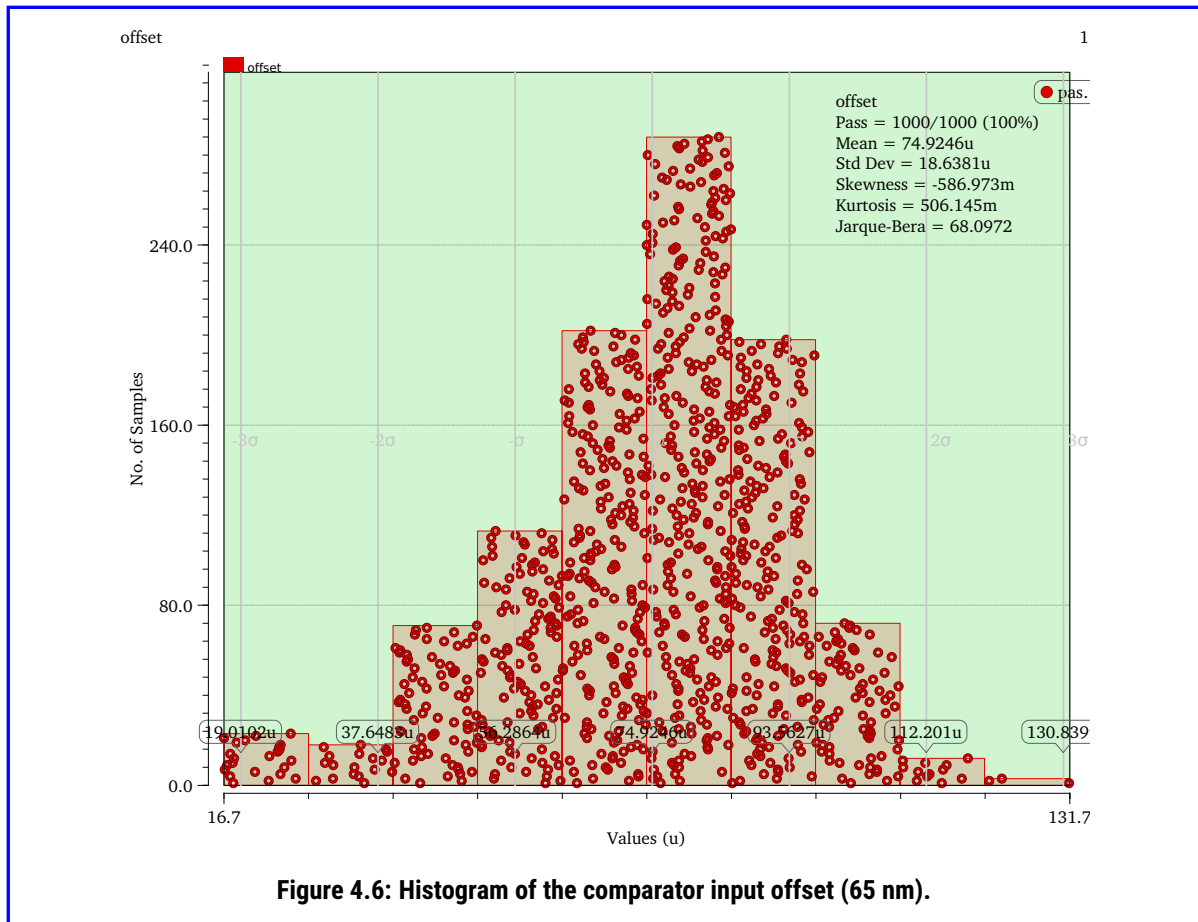


Figure 4.5 illustrates the top-level architecture of the implemented SAR ADC incorporating memristor-assisted CDAC calibration scheme. The switching scheme of the SAR logic is a Vcm-based type, which has a relatively good power efficiency compared to other switching schemes (see Figure 4.2). During the sampling phase, the differential input voltages are sampled on top plates of MOM DAC capacitors. Then in the conversion phase, the SAR logic contains 12 regular conversion cycles, 1 redundancy cycle and 1 calibration cycle. The SAR logic is based on a Vcm-based asynchronous architecture, which is similar to the one proposed in [92]. Similar to the TSPC-based SAR logic shown in Chapter 3, the asynchronous logic is exploited to ensure the calibration cycle can be successfully activated when one of the calibration activation codes is detected. This is because the internal clock signals are generated by internally triggered asynchronous logic, so each conversion cycle can only be triggered by the preceding cycle. Thus, the 14th calibration cycle can only be triggered after the 13th cycle and 'Calibration Activation' signal is activated. The calibration detection and sign generation logic are exactly the same as the one used for the RDAC calibration, so the circuits design can be directly transferred from 180 nm to 65 nm with only changing the standard cells. Detailed timing diagrams showing the proposed logic are shown in later section. This chapter mainly focuses on the implementation of the MOM CDAC and V_{ref} calibration logic.

4.3.2 Implementation of the Comparator

The comparator implementation maintains the architecture presented in Figure 3.11, with transistor dimensions and operating voltage optimized for the 65 nm process node. Monte Carlo simulation results from the testbench shown in Figure 5.2, using a 1000-sample dataset, characterize the comparator's offset distribution. For a 12-bit SAR ADC operating with a 1 V DAC reference voltage, the design requires an input offset below $0.5V_{ref}/2^{12} = 122 \text{ } \mu\text{V}$ to maintain conversion accuracy. As shown in Figure 4.6, the implemented comparator achieves a mean offset value of 74.9 μV , comfortably meeting the 3σ and design requirements.



4.3.3 Vcm-based Switching SAR Logic & Switching Control

In this design, the SAR is in a Vcm-based switching logic, following architecture similar to the one used in [92] and [93]. Different from the conventional 'Trial & Error'¹ switching logic shown in Chapter 3, in every conversion cycle of the Vcm-based switching logic, only one switching behaviour will appear, achieving a better energy efficiency compared to the conventional one. Thus, the previously designed TSPC logic cannot be re-used here. Instead, as shown in Figure 4.7, the asynchronous-clear type DFFs from the standard cell library are exploited to construct the Vcm-based SAR logic. The output clock signals are triggered one by one and applied to the switching circuits shown in Figure 4.8. The switching circuit is able to provide the V_{ref} for the DAC capacitor arrays. The switching circuit for the MSB capacitors is able to provide a compensated V_{ref} to adjust the charge redistribution process and thus calibrate the overall ADC linearity performance (Please refer to Section 4.2.).

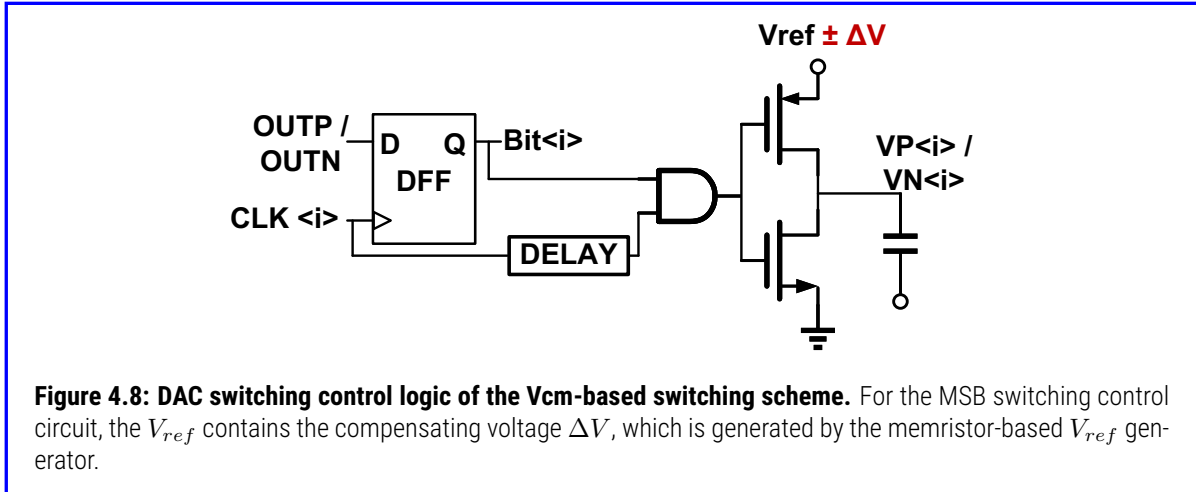
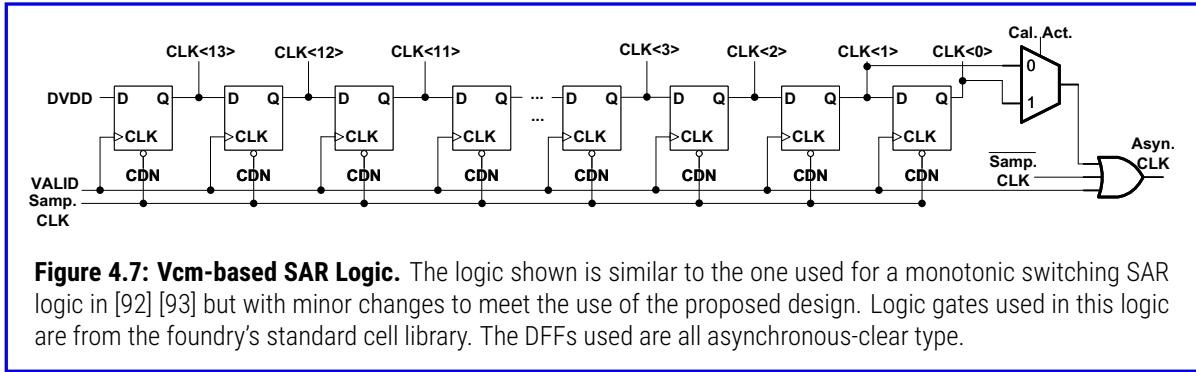
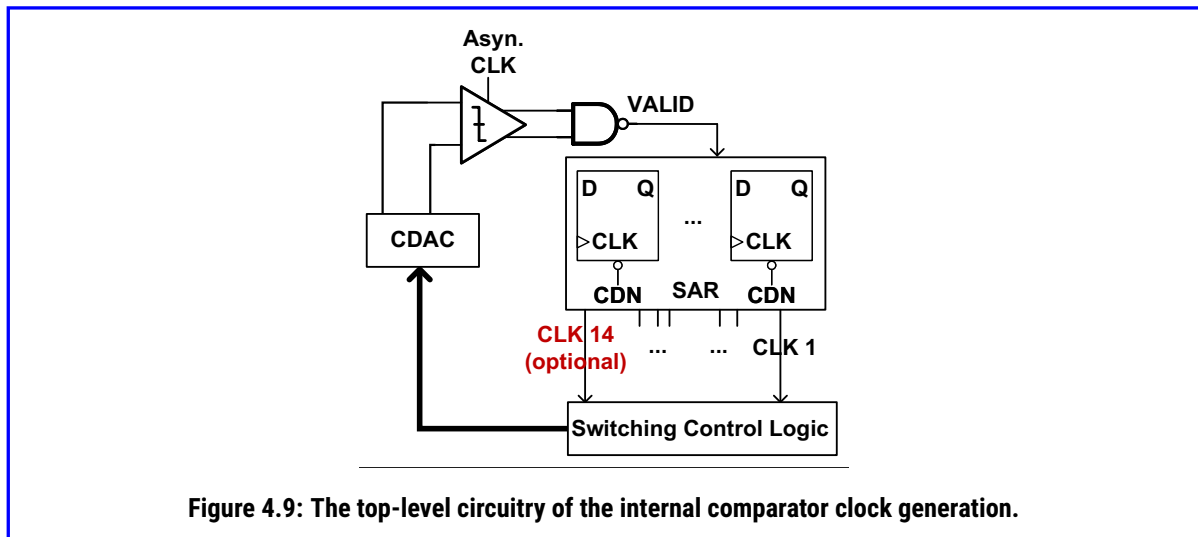


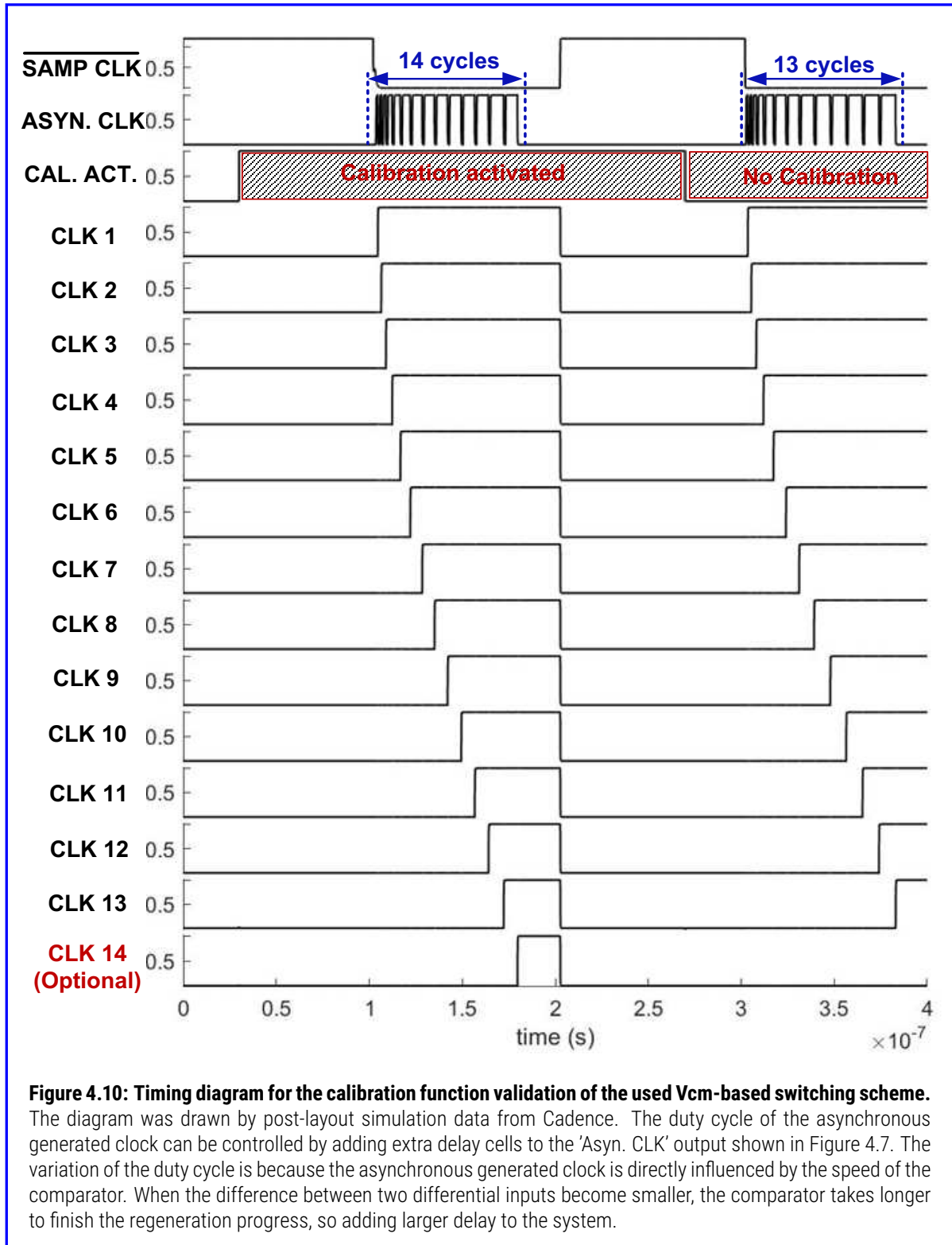
Figure 4.9 illustrates the overall architecture of the asynchronous clock generation logic. The 'VALID' signal is set to '0' when the comparator resets its output to '1', and it transitions to '1' once the comparator completes its comparison (when the two outputs of the comparator are opposite).

¹The 'Trial & Error' switching SAR logic is the most commonly used SAR logic in past decades. Before each bit conversion, the logic first sets the bit to '1' (Trial). After the comparison, if the comparator output meets the pre-set '1', the logic will keep the pre-set value, or the logic will set the bit to '0' (Error). Because of this working principal, the 'Trial & Error' switching logic has a relatively high energy consumption compared to other switching logic.



In this method, CLK<13> to CLK<0> are sequentially triggered high, enabling the internal generation of the clock signal for the comparator. To ensure one optional calibration cycle can be activated when the 'calibration activation' signal is high, a multiplexer is applied to select between signal CLK<1> and CLK<0> (see Figure 4.7). If calibration is not activated, the CLK<1> is selected and 13 clock cycles will be generated. When the calibration is activated, the CLK<0> is selected, generating one more extra conversion cycle so the comparator can perform one more comparison with altered MSB code.

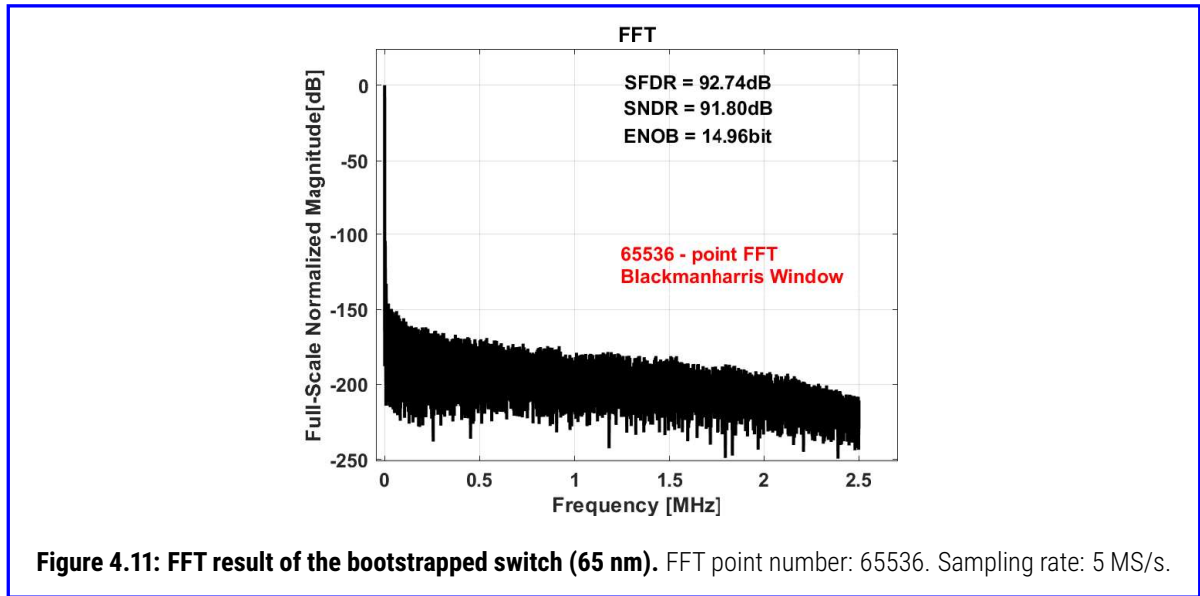
Figure 4.10 shows the timing diagram of the proposed SAR logic. When the calibration is activated, CLK<14> is triggered high and 14 clock cycles are generated. If the calibration is not activated, the CLK<14> keeps low and only 13 clock cycles are generated. The timing diagram result shows that the implemented SAR logic is able to achieve the system function proposed in Section 4.2. The duty cycle of the asynchronous generated clock can be adjusted by adjusting the number of delay cells after the 'VALID' signal.



4.3.4 Implementation of the Bootstrapped Switch

The schematic of the bootstrapped switch is same as the one shown in Figure 3.8 but with different values for capacitors. The C1 was selected as 1 pF, and C2, C3 were selected as 53 fF. The sampling switch must have enough settling time to ensure the settling error ($\geq 9.01 \frac{T_s/2}{\tau}$ for this design)² To calculate the time constant ($\tau = RC$) of the proposed RC network, the testbench shown in Appendix figure 5.1 was used to get the on-resistance (R_{on}). The holding capacitor used in this design is 1.05575 pF³, then the $T_{required} \geq \ln(2 \cdot 2^{12}) \cdot 2R_{on}C = 6.8ns$. The sampling rate of the design is far more enough for the sampled signal to settle (1/5 MHz = 200 ns). The 1.05575 pF holding capacitor can also easily meets the thermal noise requirement of the sampling network, i.e., $\frac{kT}{C} \leq \frac{V_{LSB}^2}{12}$.

To ensure a compact layout design, full-custom Metal-Oxide-Metal (MOM) capacitors have been designed for C1 - C3. Figure 4.12 shows the design method of the MOM capacitor used for the bootstrapped switches. Metal 3 and Metal 2 are stacked through Via connections to get larger capacitance value. By connecting top plates and bottom plates of individual unit cells (C_u) together, it is easy to get larger capacitance value which is suitable for this design. The capacitance value between the top plate (TOP) and the bottom plates (BOT) can be easily extracted by parasitic extraction tools (with c+cc and highest coupling capacitance extraction resolution settings) provided by SIEMENS.



²The first order RC response equation is $V_{out}(t) = V_{FS}(1 - e^{-t/\tau})$, then the settling error is $V_{out,err}(t) = -V_{FS}e^{-t/\tau}$. Assume the settling error needs to be smaller than 0.5 LSB, we can get a equation $\frac{T_s/2}{\tau} \geq \ln(2 \cdot 2^{ENOB})$.

³The holding capacitor for the CDAC-based SAR ADC is the total capacitance (C_{tot}) of the CDAC arrays. In this design, the unit capacitance (C_u) is 250 aF. More details of the DAC design is shown in Section 4.3.5.

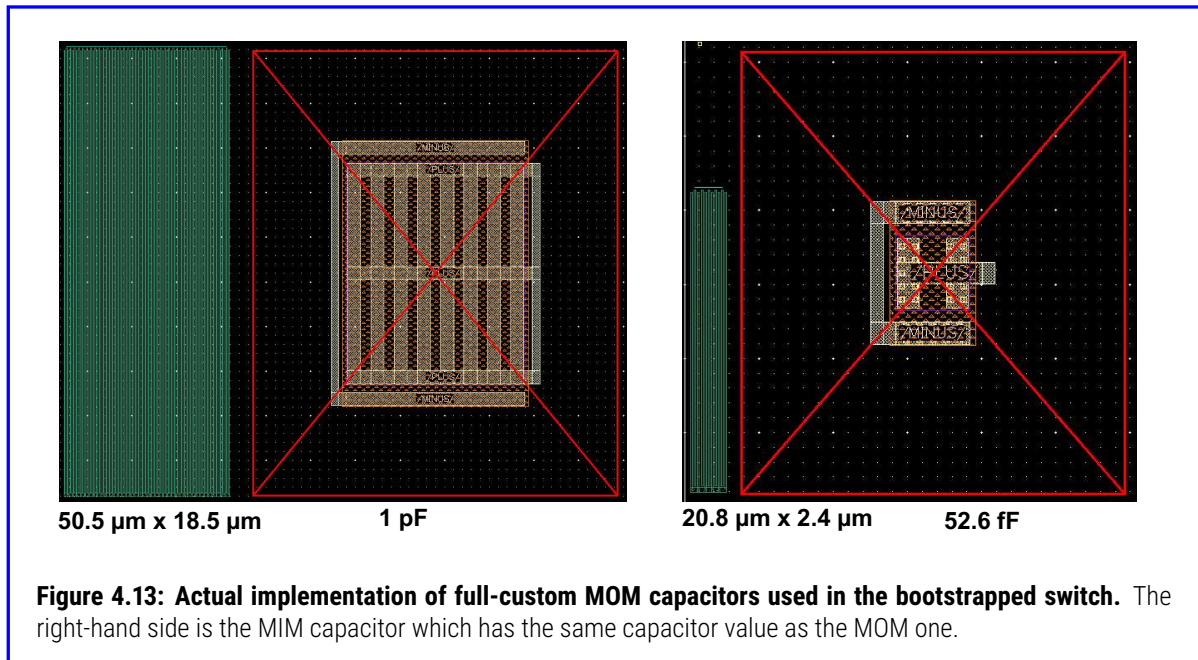
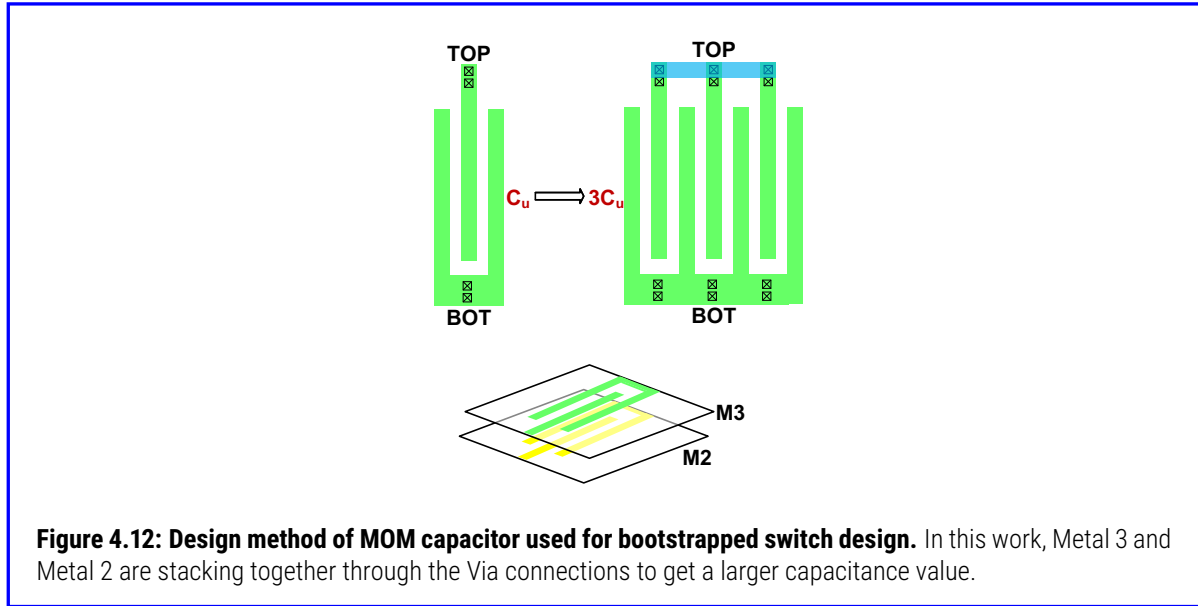


Figure 4.13 shows the actual implementation of the MOM capacitors used in the bootstrapped switch. Compared with the MIM capacitors available in the pdk, the full-custom MOM capacitors has a compact size and are able to be designed in bottom metal layers, which can be placed under the DAC arrays to pursue a compact layout (N.B., design method in [89] was used here). Figure 4.11 shows the FFT result of the proposed switch at 5 MS/s with 65536 (2^{16}) sample points. The result shows that the proposed switch can achieve an ENOB of 14.96 bits, which can meet the linearity requirement of the 12-bit ADC design. The simulation considers all non-ideal effects extracted from the layout (the full-custom MOM capacitor can only run in post-layout simulation). Figure 4.14 shows the ENOB histogram of the bootstrapped switch, which is generated from a 875-sample post-layout Monte Carlo simulation at a sample rate of 5 MS/s. These results prove that

the proposed switch is able to maintain a good input linearity for the proposed ADC design. (N.B., the frequency of the input stimulus is 992 Hz, which is selected to achieve a coherent sampling. This means an integer number of cycles inside the predefined sampling window [77]. In this case, f_{sample} is 5 MHz, $N_{window} = 13$, $N_{record} = 65536$. Thus, the $f_{In} = \frac{5MHz \times 13}{65536}$.)

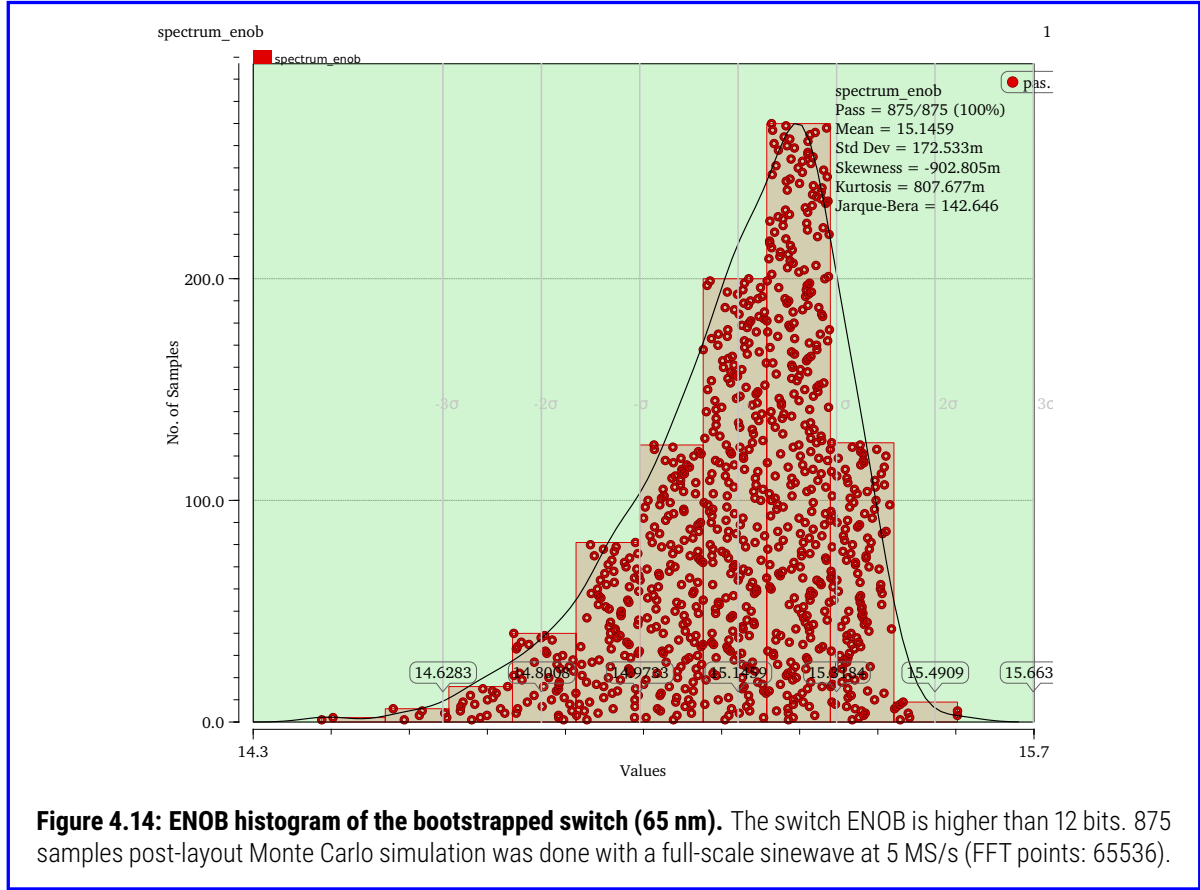
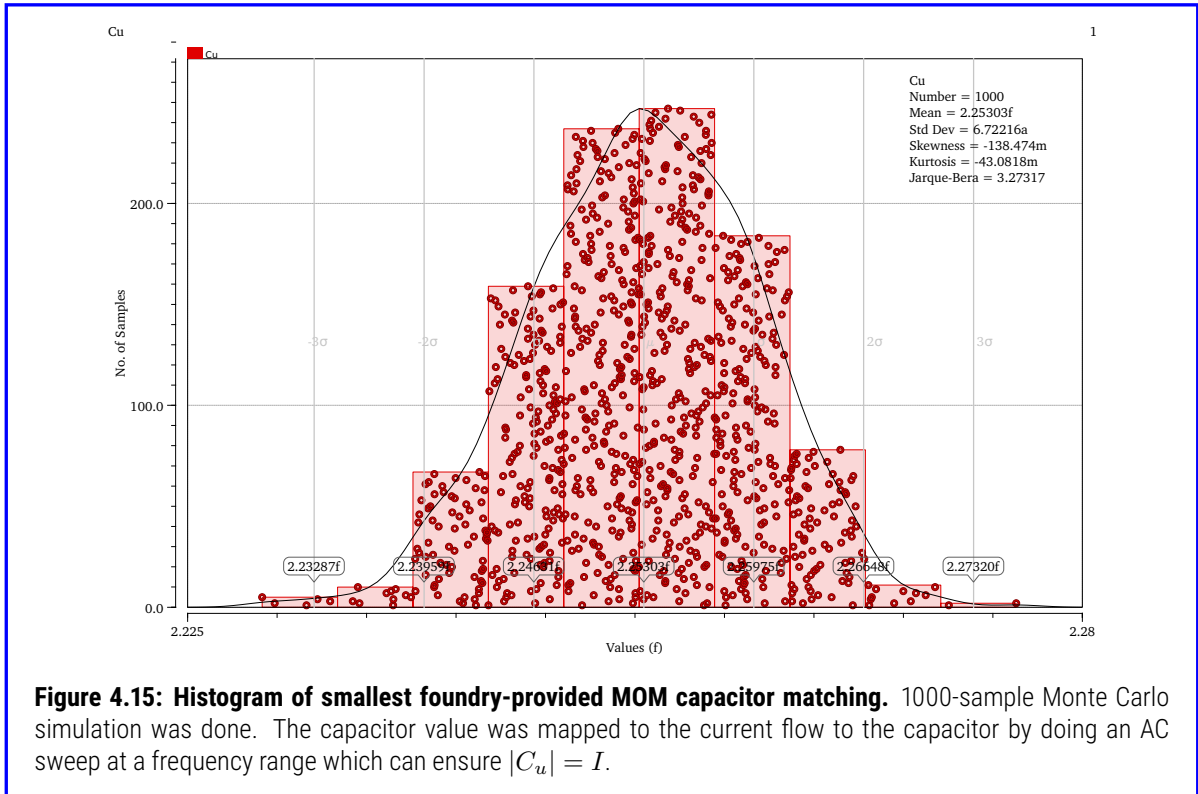


Figure 4.14: ENOB histogram of the bootstrapped switch (65 nm). The switch ENOB is higher than 12 bits. 875 samples post-layout Monte Carlo simulation was done with a full-scale sine wave at 5 MS/s (FFT points: 65536).

4.3.5 Implementation of full-custom MOM DAC arrays

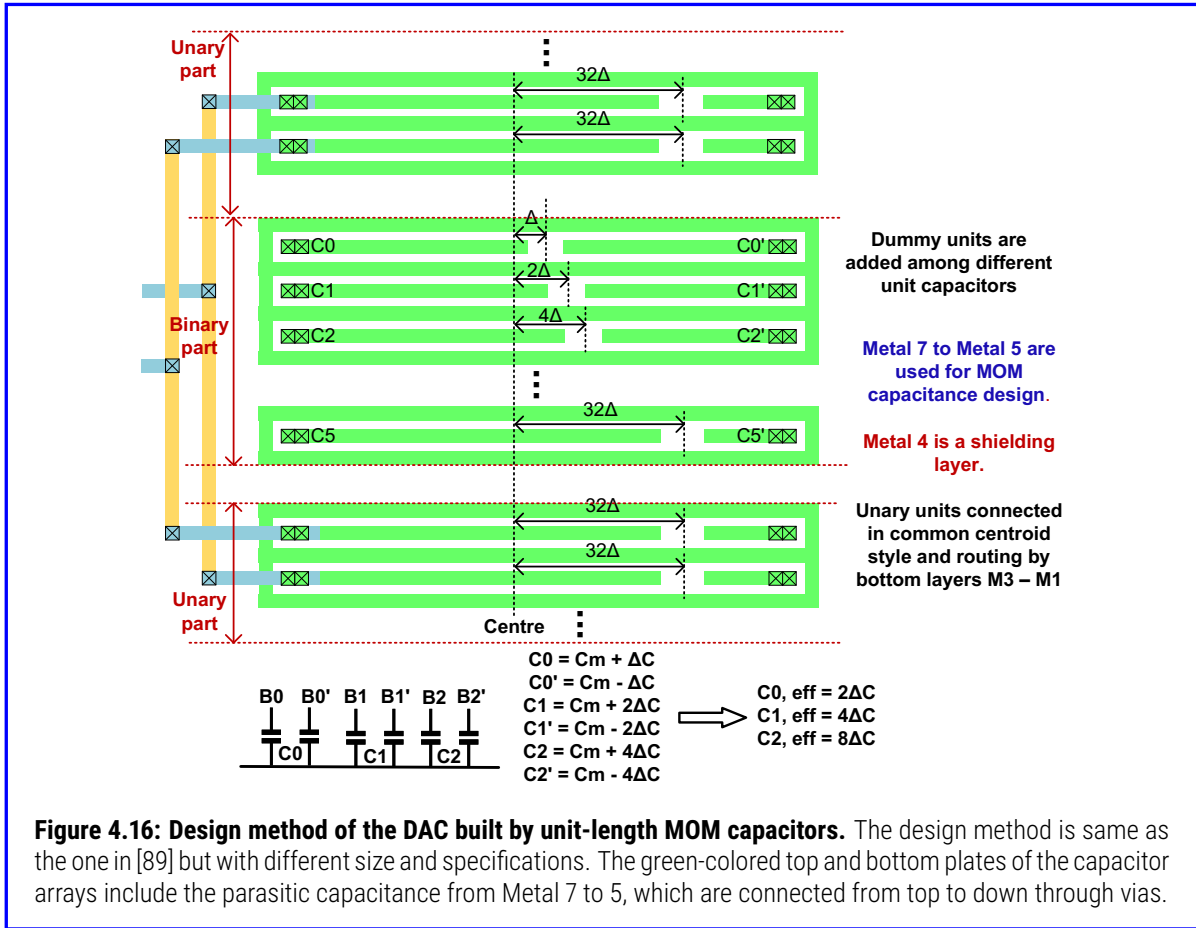
The unit capacitance (C_u) value was selected to meet the noise and the matching requirements (i.e., $\frac{kT}{C_{tot}} \leq \frac{V_{LSB}^2}{12}$, $3\sigma_{DNL,max} = 3\sqrt{2^N - 1} \frac{\sigma_u}{C_u} LSB < 0.5 LSB$). For a 12-bit ADC with a 1 V power supply and reference voltage, the $V_{LSB} = \frac{1}{2^{12}} = 0.244 mV$, so a total capacitance of $C_{tot} \geq 833 fF$ is able to meet the thermal noise requirement. In this design, one redundancy bit has been introduced, then $C_{tot} = (2048 + 1024 + 512 + 256 + 128 + 128 \cdots + 1)C_u = 4223C_u$. The C_u needs to be larger than 197 aF to meet the thermal noise requirement.

For a 12-bit analog-to-digital converter (ADC), the capacitor mismatch must be smaller than 0.26% to meet the required matching specifications. Figure 4.15 shows the matching histogram of the smallest metal-oxide-metal (MOM) capacitor provided by the foundry. Based on the data, the capacitor mismatch is calculated to be 0.30%, which does not meet the matching requirement. Additionally, the capacitance value of this foundry-provided capacitor is relatively large compared to the minimum capacitance (C_u) needed to satisfy the thermal noise requirement (197 aF). This large capacitance leads to higher power consumption and longer settling time for the ADC design.



To ensure a compact DAC design with a relatively good matching, a full-customized MOM capacitors are required. In this design, the method in [89] was exploited to design the DAC. As illustrated in Figure 4.16, the unit-length capacitor is designed by exploiting the capacitance difference between 2 capacitors sharing the same top plates (e.g., the effective capacitance of C_0 is $C_0 - C_0'$, which can be achieved by applying an opposite signals to two bottom plates respectively in each unit cell).

By extending the length of the top plate metal in a binary weighted manner, it is able to achieve a binary weighted CDAC arrays. However, due to the limit unit length, it is not able to scale the ΔV to 13th bit. To solve this issue, for higher bit's capacitors, a unary design style was used, and unit cells for one bit were placed and routed in a common centroid layout pattern (see Figure 4.16, and the complicated placing and routing can be easily done by available EDA tool embedded in Cadence).



The DAC layout was designed by using this design method, Figure 4.17 shows the layout design of two 13-bit (1 redundancy bit) DAC arrays. Metal 7 to Metal 5 were used to build the capacitors. Metal 4 is a shielding layer (ground layer), which was added to reduce the parasitic effects from the unit-length cells. Metal 3 and Metal 2 were used to do the routing for unary cells of the DAC. Figure 4.17 (d) shows the final DAC layout design, where each DAC array only occupies an area of 30.48 μm x 76.34 μm . The capacitance of each bit can be extracted by using SIEMENS calibre in c+cc mode, with xACT 3D Hybrid MEMS for a better accuracy. Table 4.2 shows the extracted capacitance value of the proposed DAC arrays. The minimum Δ is 0.24 μm , achieving a unit capacitance (C_u) of 230 aF. b0 - b5 were designed in binary-weighted scale, and b6 - b12 were designed in unary scale. b7 is the redundancy bit, so its weight (31.16 fF) repeats the one of b8 (31.14 fF). Thanks to the design method proposed in [89], overall a good binary weighted relationship can be guaranteed among all DAC bits (N.B., the extraction was done with reference to a ground node placed on Metal

4 shielding).

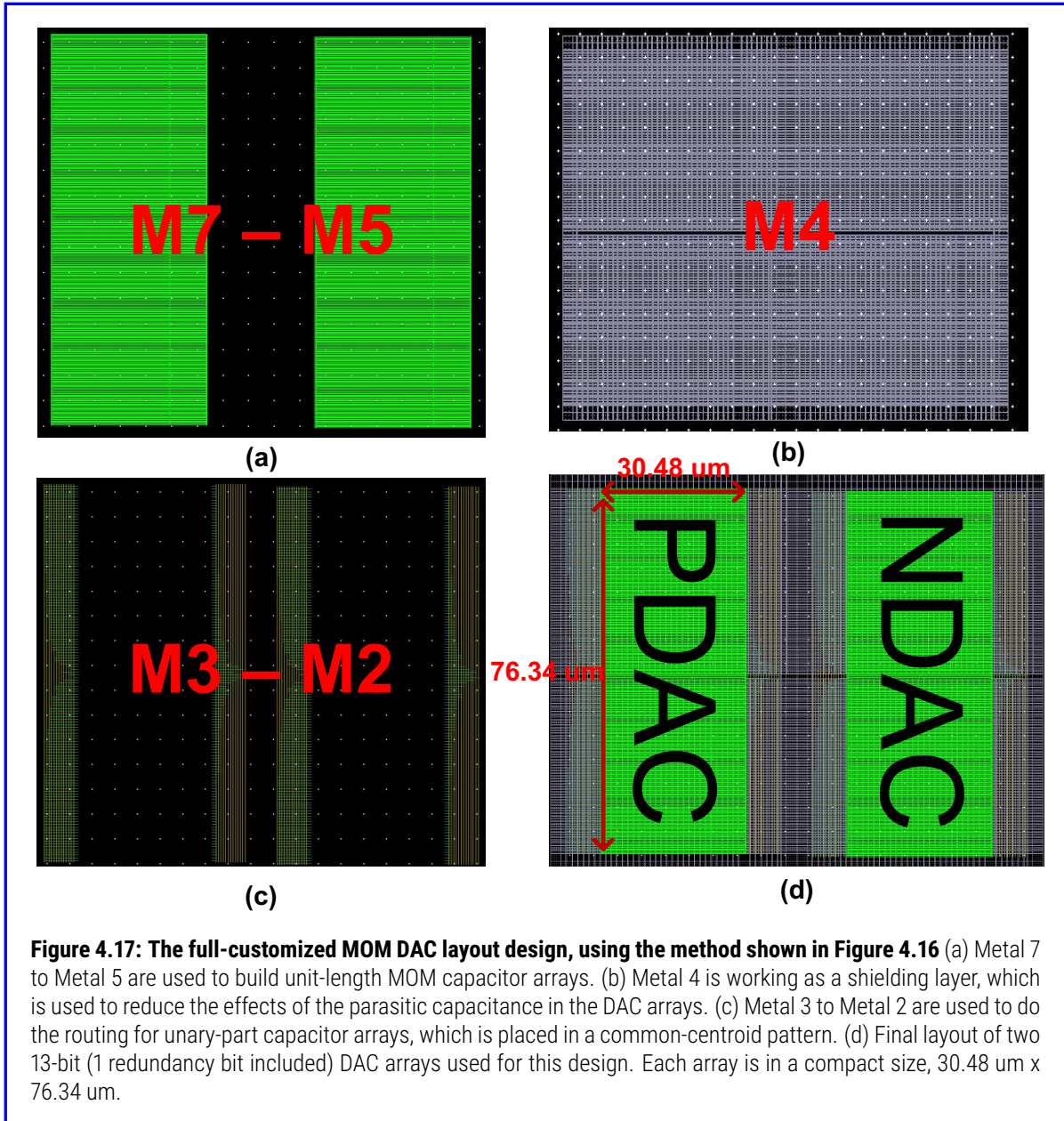
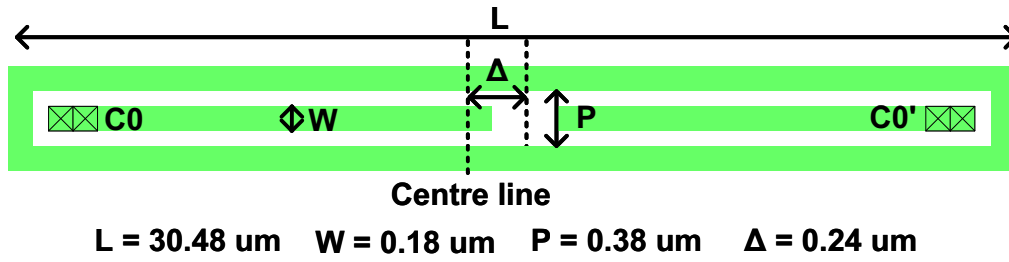


Table 4.2: MOM capacitor value extraction. (Using Siemens calibre tool with xACT 3D hybrid MEMS.)



Bit	Unit Num.	C (fF)	C'(fF)	Ceff (fF)		Binary Weight
0	1	7.92	7.69	0.23	Binary	1
1	1	7.67	7.20	0.47		2
2	1	7.91	6.96	0.95		4
3	1	8.39	6.47	1.92		8
4	1	9.81	5.78	4.03		16
5	1	11.33	3.54	7.79		32
6	2	22.64	7.08	15.16	Unary	64
7*	4	45.30	14.14	31.16		128
8	4	45.28	14.14	31.14		128
9	8	90.54	28.26	62.38		256
10	16	181.10	56.51	124.59		512
11	32	362.10	113.00	249.10		1024
12	64	723.97	225.96	498.01		2048
* Bit 7 is the redundant bit, repeating the binary weights of Bit 8						
** Extractions were done in xACT 3D Hybrid mode with MEMS precision						

4.3.6 Memrsitor-Assisted Voltage Generator

Two simple voltage dividers, each consisting of memristors, are used to provide the compensated voltage V_{refp} and V_{refn} . As shown in Fig. 4.18, programming pulses are applied to trim the memristors when mismatches are detected. In idle mode, all switches are turned off to break the paths and minimize power consumption. This work employs the Verilog-A model proposed in [39] to simulate the behaviour and characteristics of the memristors. Once the calibration direction (sign error) is determined, the programming polarity can be identified, as shown in Table 4.3.

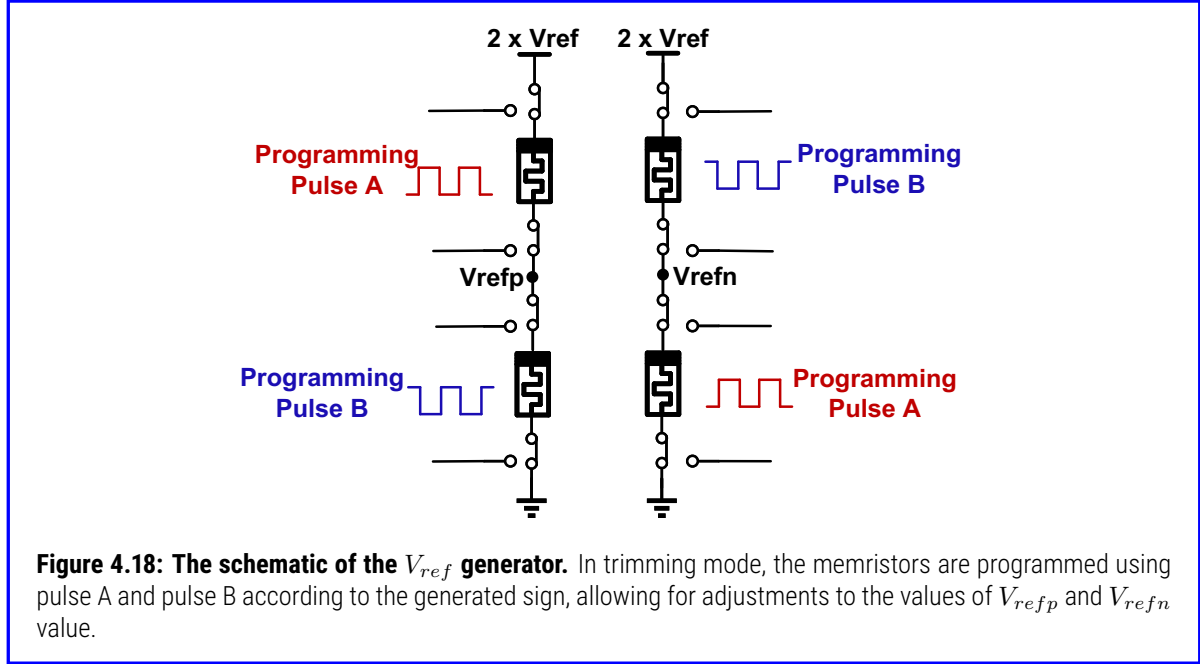


Table 4.3: Look-up table for programming polarity

Sign Error	polarity		compensating voltage	
	pulse A	pulse B	V_{refp}	V_{refn}
+ ↓	-	+	↑	↓
- ↑	+	-	↓	↑

For instance, a programming pulse A increases the memristance from $43k\Omega$ to $48k\Omega$, while the programming pulse B decreases the memristance to $38k\Omega$ respectively. Consequently, the V_{refp} becomes $2V_{ref} \times \frac{38}{38+48} = 0.88V_{ref}$. By adjusting the memristance in a step of $5k\Omega$, a 0.12 V adjustment in V_{ref} can be achieved. This method allows users to fine-tune the calibration resolution by modifying the amplitude and pulse width of the programming pulses. A key advantage of this correction technique is its ability to operate in the background without incurring additional conversion time, as V_{ref} generation and calibration are independent of the SAR ADC's timing logic. This significantly enhances the SAR ADC speed compared to the calibration method discussed in the previous chapter.

4.4 Floorplan of Proposed Design

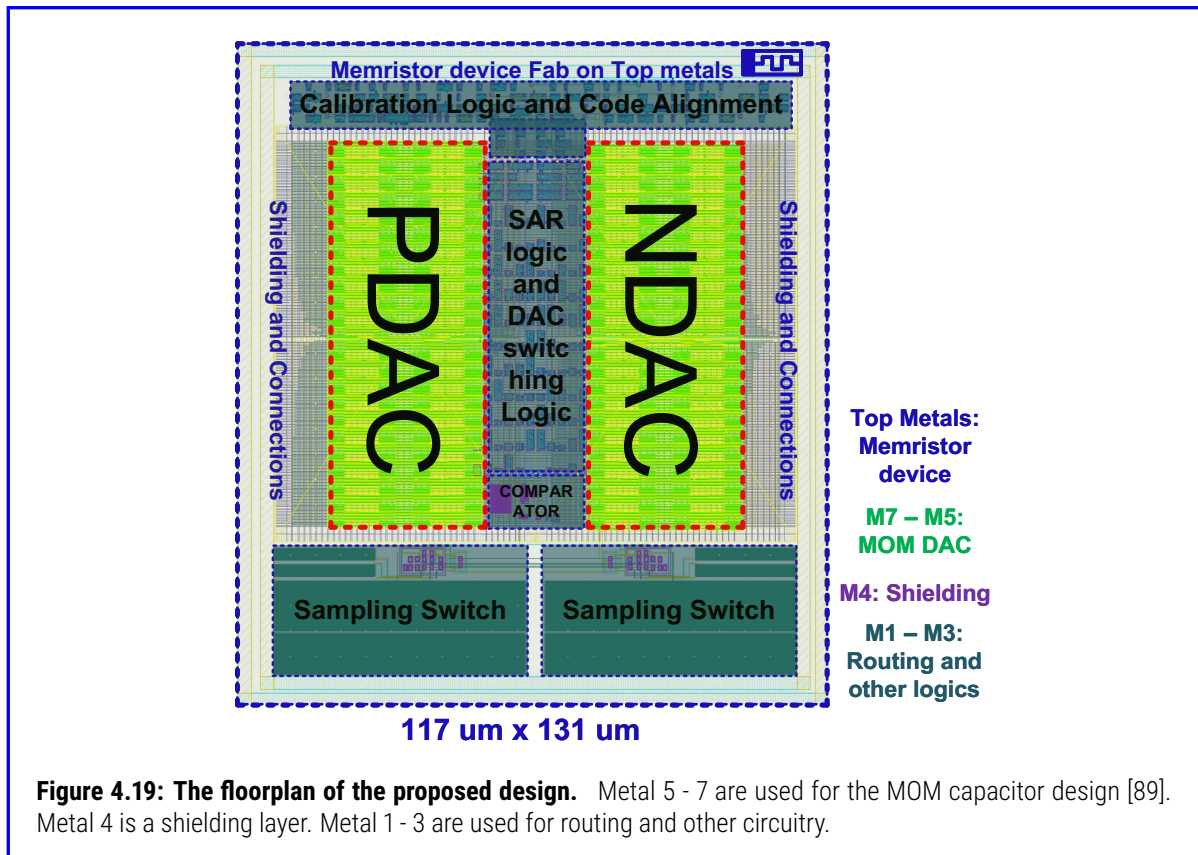
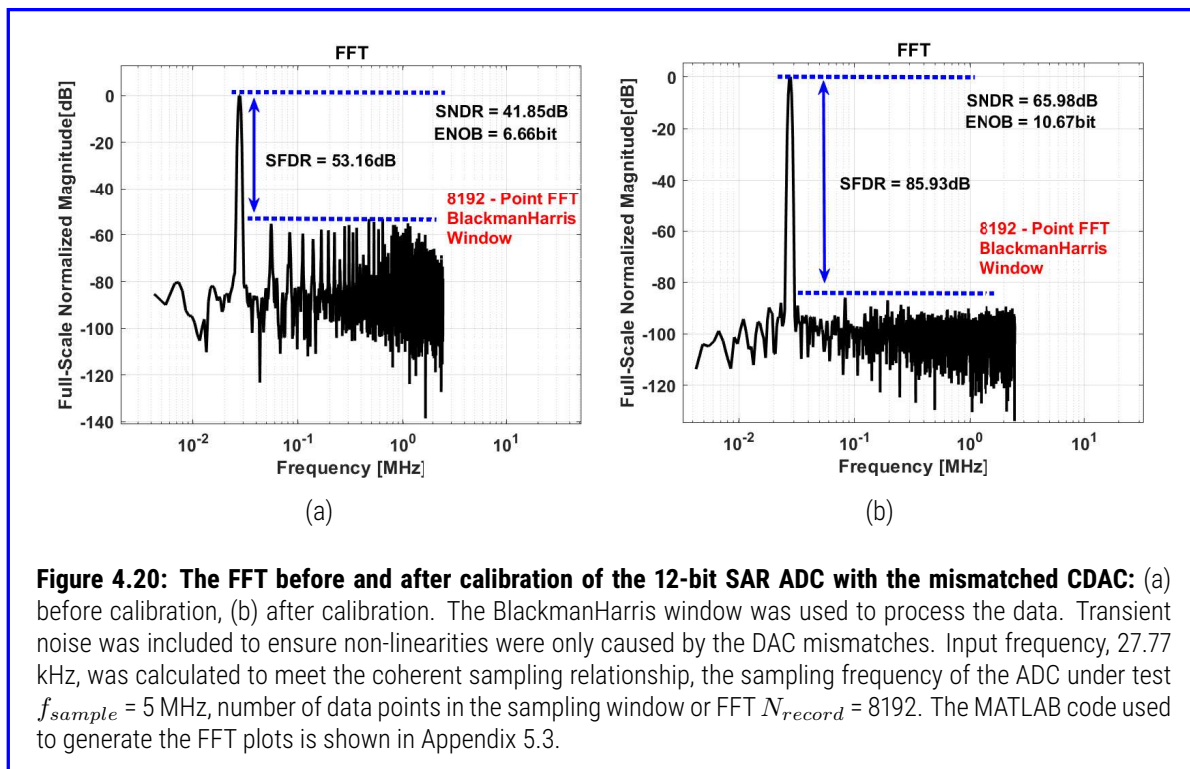


Figure 4.19 presents the floorplan of the ADC core design, which demonstrates an optimized layout architecture utilizing a seven-metal-layer process. The design achieves a good area efficiency through a full-customized MOM DAC implementation, resulting in a compact layout measuring (0.0153 mm^2). The floorplan comprises three distinct sections: the PDAC and NDAC arrays implementing the primary conversion elements, centrally positioned SAR logic and DAC switching circuitry, and two bootstrapped sampling switches were placed at the bottom of the layout.

The metal stack is efficiently organized to maximize performance and minimize the area. Metals 5 through 7 are dedicated to the MOM capacitor design [89], maintaining compact dimensions. Metal 4 serves as a dedicated shielding layer, crucial for minimizing parasitic effects. The lower metal layers (Metals 1-3) are utilized for routing and peripheral logic implementation.

A feature of this architecture is the integration of memristor devices on the top metal layers (M8, M9 are available in the standard 65 nm process). This approach enables the implementation of calibration functionality without incurring additional silicon area overhead. The placement of memristors in the top metal layers represents an elegant solution to area constraints while maintaining the design's calibration capabilities. This floorplan demonstrates careful consideration of both area efficiency and signal integrity, resulting in a compact yet robust design suitable for high-performance mixed-signal applications.

4.5 Simulated FFT result of the Proposed SAR ADC



Fast Fourier Transform (FFT) analyses were performed to evaluate the performance of the 12-bit ADC at a sampling rate of 5 MS/s with a 27.77 kHz sine-wave input signal. Figure 4.20 presents the FFT spectrum both before and after the implementation of the calibration scheme. The uncalibrated ADC exhibits a SNDR of 41.85 dB, a SFDR of 53.16 dB, and an ENOB of 6.66 bits. After applying the proposed calibration method, these performance metrics were effectively improved, achieving an SNDR of 65.98 dB, SFDR of 85.93 dB, and ENOB of 10.67 bits, representing improvements of 24.13 dB, 32.77 dB, and 4.07 bits, respectively. The FFT analysis was conducted using a BlackmanHarris window with 8192 data points to ensure accurate spectral characterization. In the pre-calibration measurements, significant harmonic distortion was observed, primarily attributed to capacitor mismatch in the DAC arrays. The proposed calibration scheme effectively addresses these non-idealities, as demonstrated by the marked reduction in harmonic components across the frequency spectrum. The calibration methodology successfully mitigates the effects of capacitor mismatch, so suppressing the harmonic distortions, resulting in a better dynamic linearity performance (Note: The simulation only considers the parasitic effects from the DAC arrays and bootstrapped switches. The FFT result shown here demonstrates the best case ADC performance compared to results got with other test signals.).

4.6 Discussion & Conclusions

Table 4.4: Performance comparison

	JSSC'17 [59]	TCAS I'23 [2]	This work [84]
Architecture	SAR ADC with sign-based calibration		
Technology (nm)	40	180	65
Area (mm^2)	0.0675	0.0531*	0.0153
ENOB (bit)	10.40	10.28**	10.67
Supply voltage (V)	1.0	1.8	1.0
Sample rate (MS/s)	6.4	0.1	5.0
SNDR (dB)	64.10	63.62**	65.96
SFDR (dB)	81.90	80.87**	85.93

* Counting 1 extra DAC for testing.

** Results obtained from additional simulations.

Method in [59] is aligned to a US patent [85], and used in [86–88].

This work proposed a memristor-assisted CDAC background calibration technique for a 12-bit SAR ADC, based on a stacked $Pt/TiO_x/Al_xO_y/Pt$ memristor model. System-level simulations demonstrate that this calibration technique effectively suppresses harmonic distortion in the 12-bit SAR ADC with CDACs, achieving significant improvements in performance metrics. The SNDR and SFDR were enhanced by 24.13 dB and 32.77 dB respectively, while ENOB increased by 4.07 bits, establishing the feasibility of using memristors in CDAC calibration for SAR ADCs.

This approach successfully addresses several limitations found in previous memristor-assisted calibration for SAR ADCs with RDACs, providing a more compact solution as evidenced in Table 4.4. The implemented design achieves a notable reduction in area (0.0153 mm^2) while maintaining competitive performance metrics. The transition from RDAC to CDAC architecture, combined with the memristor-assisted calibration scheme, demonstrates the potential for advancing ADC design, particularly for applications requiring both high accuracy and compact size, which is suitable for compact and low-power array-based interface applications.

However, several aspects require further exploration to advance this research. While the current work provides system-level validation, a comprehensive chip-level characterization remains essential for fully validating the proposed calibration scheme. The parasitic effects and non-idealities of memristors, such as device-to-device and cycle-to-cycle variations, need to be thoroughly characterized in the model proposed in [39], through physical implementation and testing. The present simulation-based implementation, though incorporating post-layout parasitic effects of the DAC arrays and bootstrapped switches, cannot fully capture all real-world circuit interactions and device behaviors. Additionally, the design primarily focuses on MSB calibration, leaving room for expansion to address LSB mismatch effects. A complete chip-level validation would also enable comprehensive investigation of temperature effects and long-term stability of the calibration scheme,

aspects that are crucial for practical applications. Thus, the future work is needed to further verify the efficacy of the proposed scheme. Future research directions should focus on:

- Characterization of memristor device-to-device, cycle-to-cycle and noise behaviours under various operating conditions (i.e., corners and Monte Carlo models)
- Extension of the calibration scheme to include LSB mismatch
- Investigation of reliability and aging effects

Nonetheless, this work represents a valuable contribution to both the ADC and emerging device communities, offering a foundation for more in-depth studies as memristor technology continues to advance. The successful demonstration of improved performance metrics while achieving substantial area reduction validates the potential of memristor-assisted calibration in next-generation SAR ADC designs.

Chapter 5

Conclusions & Future Prospects

This chapter opens with a concise summary of the thesis's primary findings and key contributions in advancing memristor-assisted ADC calibration techniques. Following this synthesis of the research outcomes, a comprehensive list of publications is presented, documenting the work's dissemination within the Circuits and Systems (CAS) community. The chapter then critically examines the limitations and technical challenges encountered during the investigation of memristor-assisted ADC calibration techniques. Through this analysis, several crucial areas requiring further development are identified, providing clear directions for future research. The chapter concludes with a forward-looking discussion on the prospects of ADC design enhanced by emerging devices, considering both the current technological landscape and anticipated developments in the field. This perspective offers valuable insights into the potential evolution of mixed-signal circuit design as emerging technologies mature.

5.1 Conclusions

This thesis has demonstrated the feasibility and potential of memristor-assisted background calibration in high-resolution ADCs through two distinct implementations. The first approach, utilizing a resistive DAC architecture, established the fundamental concept of exploiting memristors' non-volatile resistance switching properties for calibration. By integrating a memristor within the RDAC's MSB branch, the system achieved significant performance improvements, with SNDR enhanced by 14.26 dB and SFDR improved by 21.11 dB. This implementation validated the core premise that memristors could effectively serve as both calibration memory and trimming elements.

Building upon these findings, the second implementation addressed the limitations of the RDAC approach by developing a voltage-compensated calibration scheme for CDAC-based architectures. This enhanced design not only improved performance metrics substantially - achieving a 24.13 dB improvement in SNDR and 32.77 dB enhancement in SFDR - but also demonstrated remarkable efficiency in silicon area utilization, reducing the footprint by approximately 4.5 times compared to previous implementations. The transition from direct resistance trimming to voltage compensation proved particularly effective in maintaining calibration accuracy while minimizing the impact of memristor variability. These results, while preliminary in nature, represent a significant step forward in the integration of emerging devices with traditional mixed-signal circuits. The successful demonstration of memristor-assisted calibration opens new possibilities for addressing fundamental limitations in ADC design, particularly in the context of element matching and area efficiency.

5.2 Publications & Contributions

Contents presents in this thesis are from the following publications:

1. **Z. Si** et al., "Memristor-Assisted Background Calibration for Digital-to-Time Converters," IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 70, no. 9, pp. 3497–3508, Sept. 2023, doi: 10.1109/TCSI.2023.3272609. [2]
2. C. Wang, **Z. Si** et al., "Multi-State Memristors and Their Applications: An Overview," in IEEE Journal on Emerging and Selected Topics in Circuits and Systems, vol. 12, no. 4, pp. 723-734, Dec. 2022, doi: 10.1109/JETCAS.2022.3223295. [3]
3. **Z. Si** et al., "Memristor-Assisted CDAC Background Calibration Scheme for SAR ADCs," 2025 IEEE International Symposium on Circuits and Systems (ISCAS), London, UK, 20205. [84]
4. **Z. Si** et al., "Memristor-assisted Background Calibration for Analog-to-Digital Converter," 2022 20th IEEE Interregional NEWCAS Conference (NEWCAS), Quebec City, QC, Canada, 2022, pp. 470-474, doi: 10.1109/NEWCAS52662.2022.9842108. [61]
5. G. Huang, C. Wang, **Z. Si** et al., "An Improved Data-Driven Memristor Model Accounting for Sequences Stimulus Features," 2023 IEEE International Symposium on Circuits and Systems (ISCAS), Monterey, CA, USA, 2023, pp. 1-5, doi: 10.1109/ISCAS46773.2023.10182200. [94]

The research presented in this thesis has made several fundamental contributions to both the ADC and emerging device communities, as documented through peer-reviewed publications. The initial concept and feasibility study of memristor-assisted background calibration was first presented in "Memristor-Assisted Background Calibration for SAR ADCs: A Feasibility Study" (IEEE Transactions on Circuits and Systems I, 2023) [2], establishing the theoretical framework and practical implementation considerations for integrating memristors in ADC calibration schemes. This scheme was then explored in a SAR ADC in differential CDAC architecture, further improving the overall ADC performance, published in IEEE International Symposium on Circuits and Systems (2025) [84].

The investigation of multi-state memristor characteristics and their applications, published in IEEE Journal on Emerging and Selected Topics in Circuits and Systems (2022) [3], provided crucial insights into the behavior and limitations of memristor devices in mixed-signal applications. This work laid the foundation for understanding how device-level characteristics influence system-level performance in calibration applications.

Additional contributions include the development of improved memristor modeling techniques and their validation in circuit-level simulations, as presented at the IEEE International Symposium on Circuits and Systems (2023) [94]. These modeling advances were essential for accurate prediction of calibration performance and helped bridge the gap between device physics and circuit implementation.

5.3 Limitations & Challenges

The research in memristor-assisted ADC calibration has revealed several significant technical challenges that require further investigation and development. The primary limitations stem from both device-level characteristics and system integration complexities. At the device level, the most pressing concern lies in memristor programming reliability and consistency. Current memristor devices exhibit substantial variations in switching behavior, manifesting as both device-to-device and cycle-to-cycle variations. While the feedback-based calibration approach developed in this work can partially compensate for these variations, the underlying device variability ultimately constrains the precision achievable in trimming operations. The low yield rates in current memristor fabrication processes present additional challenges for commercial viability, necessitating improvements in manufacturing technology.

The measurement challenges encountered during chip testing, as detailed in Section 3.7, further highlight the practical implications of these device limitations. Extended fabrication cycles for on-chip memristors exceeded feasible research timelines, necessitating the use of off-chip standalone devices. This compromise, combined with the low yield rates of available laboratory devices and significant deviations from VerilogA model predictions, impeded the acquisition of empirical performance data comparable to post-layout simulation results.

Retention characteristics of memristive states present another significant challenge. The resistance states can exhibit drift over time, potentially requiring periodic recalibration to maintain accuracy. While the background nature of the proposed calibration scheme helps address this issue by enabling continuous adjustment, the long-term stability implications for practical applications remain a concern. This is particularly relevant in array-based interface applications where consistent performance is crucial.

Integration challenges persist in the context of standard CMOS processes. The post-CMOS processing required for memristor integration introduces additional complexity and potential yield implications. The research was constrained by limited memristor model availability, particularly regarding process variation and stochastic behavior modeling. Programming speed limitations also impact maximum achievable sampling rates, as evidenced in the RDAC implementation where calibration overhead restricted operation to 100 kS/s.

These limitations, while significant, should be viewed in the context of rapidly evolving memristor technology. As manufacturing processes mature and modeling accuracy improves, many of these challenges may be addressable through continued research and development. Future measurements with enhanced validity are anticipated as memristor fabrication technology advances and manufacturing efficiency improves.

5.4 Further Work & Future Prospects

The field of memristor-assisted ADC calibration presents significant opportunities for both technical advancement and architectural innovation. Near-term research priorities include extending the current calibration methodology to address multiple bit positions in the DAC. While the present work demonstrates effective MSB calibration, expanding to LSB positions could provide more comprehensive non-linearity correction. This extension requires careful consideration of the cumulative effects of memristor programming on overall ADC linearity and the development of sophisticated algorithms to manage multiple calibration cycles efficiently.

A particularly promising direction lies in applying memristor-assisted calibration to Delta-Sigma Modulators (DSMs) with multi-bit quantizers. As illustrated in Appendix figure 5.3, the fundamental limitation in these architectures stems from DAC non-linearity in the feedback path. The system's transfer function reveals that while the input-to-output ratio (Y/U) approaches unity with large loop gain $H(z)$, the error-to-output ratio (Y/E_2) directly reflects DAC non-linearity. This relationship indicates that DAC errors are not noise-shaped, directly impacting the modulator's performance. Conventional solutions such as Data Weighted Averaging (DWA) address this issue through dynamic element matching (DEM) but incur significant overhead in power consumption and circuit complexity due to their sophisticated switching control logic and memory requirements. The voltage-compensated calibration approach developed in Chapter 4 could provide a promising alternative, where dynamic adjustment of the feedback DAC's reference voltage using memristor-based trimming could compensate for non-linearity without introducing additional noise sources or complex switching schemes.

The ongoing advancement in memristor technology, driven by compute-in-memory (CIM) applications, promises improved device characteristics crucial for these applications. This includes better switching uniformity, reduced variation, and enhanced retention characteristics. Recent developments in multi-state memristor technology, particularly in metal-oxide bi-layer structures, demonstrate the potential for precise resistance control necessary for high-resolution calibration. The characterization platform developed for memristor devices also plays a crucial role in understanding and modeling device behavior, as demonstrated by the ArC ONE platform [40] used in this research.

From an architectural perspective, future work should explore innovative hybrid calibration schemes that combine memristor-assisted analog trimming with digital correction techniques. This could leverage the strengths of both approaches while mitigating their respective limitations. The integration of memristor arrays could enable parallel calibration of multiple elements, potentially addressing current speed limitations in the calibration process. The compact layout achieved in the CDAC implementation suggests the possibility of even more efficient designs as process technology advances.

As memristor technology matures and integration challenges are addressed, the proposed calibration techniques could extend beyond individual ADC applications to entire mixed-signal systems. The ability to perform local, non-volatile calibration without dedicated digital circuitry could prove particularly valuable in array-based applications where area and power efficiency are important. The voltage-compensated approach demonstrated in Chapter 4 could be adapted for various mixed-signal circuits requiring precise analog trimming.

Future research should also focus on characterizing long-term reliability and developing robust models for process variation. The current limitations in statistical modeling, particularly regarding device-to-device and cycle-to-cycle variations, need to be addressed through comprehensive characterization and model development. This work could build upon the VerilogA model used in this research, extending it to include more detailed statistical behavior and aging effects.

This research has established a foundation for integrating emerging devices in mixed-signal calibration. As technology continues to advance, particularly with the strategic focus on compute-in-memory devices highlighted in recent US national initiatives [1], the principles and techniques developed here will serve as building blocks for future innovations in high-performance analog-to-digital conversion. The convergence of improved device technology with the calibration techniques developed in this thesis could enable more efficient and higher-performance ADC implementations, contributing to the broader evolution of mixed-signal circuit design in the era of emerging devices.

References

- [1] NATIONAL STRATEGY ON MICROELECTRONICS RESEARCH, NATIONAL SCIENCE AND TECHNOLOGY COUNCIL, The White House, USA. 2024. URL: <https://www.whitehouse.gov/wp-content/uploads/2024/03/National-Strategy-on-Microelectronics-Research-March-2024.pdf> (visited on 05/15/2024).
- [2] Si, Z., Wang, C., Jiang, X., et al. "Memristor-Assisted Background Calibration for SAR ADCs: A Feasibility Study". In: *IEEE Transactions on Circuits and Systems I: Regular Papers* 70.9 (2023), pp. 3497–3508. doi: 10.1109/TCSI.2023.3272609.
- [3] Wang, C., Si, Z., Jiang, X., et al. "Multi-State Memristors and Their Applications: An Overview". In: *IEEE Journal on Emerging and Selected Topics in Circuits and Systems* 12.4 (2022), pp. 723–734. doi: 10.1109/JETCAS.2022.3223295.
- [4] Chua, L. "Memristor-The missing circuit element". In: *IEEE Transactions on Circuit Theory* 18.5 (1971), pp. 507–519. doi: 10.1109/TCT.1971.1083337.
- [5] Strukov, D. B., Snider, G. S., Stewart, D. R., et al. "The missing memristor found". In: *Nature* 453.7191 (2008), pp. 80–83. ISSN: 0028-0836. doi: 10.1038/nature06932.
- [6] Pi, S., Li, C., Jiang, H., et al. "Memristor crossbar arrays with 6-nm half-pitch and 2-nm critical dimension". In: *Nature nanotechnology* 14.1 (2019), pp. 35–39.
- [7] Holloway, P. and Norton, M. "A high yield, second generation 10-bit monolithic DAC". In: *1976 IEEE International Solid-State Circuits Conference. Digest of Technical Papers*. Vol. XIX. 1976, pp. 106–107. doi: 10.1109/ISSCC.1976.1155521.
- [8] McGlinchey, G. "A monolithic 12b 3 μ s ADC". In: *1982 IEEE International Solid-State Circuits Conference. Digest of Technical Papers*. Vol. XXV. 1982, pp. 80–81. doi: 10.1109/ISSCC.1982.1156347.
- [9] Murmann, B. *ADC Performance Survey 1997-2023*. [Online]. Available: <https://github.com/bmurmman/ADC-survey>.
- [10] Oxley, D. "Electroforming, switching and memory effects in oxide thin films". In: *Electrocomponent Science and Technology* 3.4 (1977), pp. 217–224.
- [11] Pagnia, H. and Sotnik, N. "Bistable switching in electroformed metal–insulator–metal devices". In: *physica status solidi (a)* 108.1 (1988), pp. 11–65.
- [12] Asamitsu, A., Tomioka, Y., Kuwahara, H., et al. "Current switching of resistive states in magnetoresistive manganites". In: *Nature* 388.6637 (1997), pp. 50–52.
- [13] Chudnovskii, F., Odyneys, L., Pergament, A., et al. "Electroforming and switching in oxides of transition metals: The role of metal–insulator transition in the switching mechanism". In: *Journal of Solid State Chemistry* 122.1 (1996), pp. 95–99.
- [14] Beck, A., Bednorz, J., Gerber, C., et al. "Reproducible switching effect in thin oxide films for memory applications". In: *Applied Physics Letters* 77.1 (2000), pp. 139–141.
- [15] Choi, B., Jeong, D., Kim, S., et al. "Resistive switching mechanism of TiO₂ thin films grown by atomic-layer deposition". In: *Journal of applied physics* 98.3 (2005), p. 033715.
- [16] Kim, D., Seo, S., Ahn, S., et al. "Electrical observations of filamentary conduction for the resistive memory switching in NiO films". In: *Applied physics letters* 88.20 (2006), p. 202102.
- [17] Waser, R. and Aono, M. "Nanoionics-based resistive switching memories". In: *Nature Materials* 6.11 (2007), pp. 833–840.

- [18] Kim, K. M., Choi, B. J., Shin, Y. C., et al. "Anode-interface localized filamentary mechanism in resistive switching of TiO₂ thin films". In: *Applied physics letters* 91.1 (2007), p. 012907.
- [19] Gambuzza, L. V., Frasca, M., Fortuna, L., et al. "Memristor crossbar for adaptive synchronization". In: *IEEE Transactions on Circuits and Systems I: Regular Papers* 64.8 (2017), pp. 2124–2133.
- [20] Gomez, J., Vourkas, I., Abusleme, A., et al. "Voltage divider for self-limited analog state programming of memristors". In: *IEEE Transactions on Circuits and Systems II: Express Briefs* 67.4 (2019), pp. 620–624.
- [21] Fyrigos, I.-A., Ntinis, V., Vasileiadis, N., et al. "Memristor Crossbar Arrays Performing Quantum Algorithms". In: *IEEE Transactions on Circuits and Systems I: Regular Papers* (2021).
- [22] Weiher, M., Herzig, M., Tetzlaff, R., et al. "Improved vertex coloring with NbO memristor-based oscillatory networks". In: *IEEE Transactions on Circuits and Systems I: Regular Papers* 68.5 (2021), pp. 2082–2095.
- [23] Ascoli, A., Demirkol, A. S., Tetzlaff, R., et al. "Edge of Chaos Theory Resolves Smale Paradox". In: *IEEE Transactions on Circuits and Systems I: Regular Papers* (2022).
- [24] Borghetti, J., Snider, G. S., Kuekes, P. J., et al. "'Memristive' switches enable 'stateful' logic operations via material implication". In: *Nature* 464.7290 (2010), pp. 873–876.
- [25] Kim, D., Yu, C., Xie, S., et al. "An Overview of Processing-in-Memory Circuits for Artificial Intelligence and Machine Learning". In: *IEEE Journal on Emerging and Selected Topics in Circuits and Systems* (2022).
- [26] Cheng, H., Hsieh, E., Huang, Z., et al. "A novel rewritable one-time-programming OTP (RW-OTP) realized by dielectric-fuse RRAM devices featuring ultra-high reliable retention and good endurance for embedded applications". In: *2018 International Symposium on VLSI Technology, Systems and Application (VLSI-TSA)*. IEEE. 2018, pp. 1–2.
- [27] Yao, P., Wu, H., Gao, B., et al. "Fully hardware-implemented memristor convolutional neural network". In: *Nature* 577.7792 (2020), pp. 641–646.
- [28] Chen, W.-H., Li, K.-X., Lin, W.-Y., et al. "A 65nm 1Mb nonvolatile computing-in-memory ReRAM macro with sub-16ns multiply-and-accumulate for binary DNN AI edge processors". In: *2018 IEEE International Solid-State Circuits Conference-(ISSCC)*. IEEE. 2018, pp. 494–496.
- [29] Wu, T. F., Le, B. Q., Radway, R., et al. "A 43pJ/Cycle non-volatile microcontroller with 4.7 μ s shutdown/wake-up integrating 2.3-bit/cell resistive RAM and resilience techniques". In: *2019 IEEE International Solid-State Circuits Conference-(ISSCC)*. IEEE. 2019, pp. 226–228.
- [30] Xue, C.-X., Chen, W.-H., Liu, J.-S., et al. "A 1Mb multibit ReRAM computing-in-memory macro with 14.6 ns parallel MAC computing time for CNN based AI edge processors". In: *2019 IEEE International Solid-State Circuits Conference-(ISSCC)*. IEEE. 2019, pp. 388–390.
- [31] Liu, Q., Gao, B., Yao, P., et al. "A fully integrated analog ReRAM based 78.4 TOPS/W compute-in-memory chip with fully parallel MAC computing". In: *2020 IEEE International Solid-State Circuits Conference-(ISSCC)*. IEEE. 2020, pp. 500–502.
- [32] Wan, W., Kubendran, R., Eryilmaz, S. B., et al. "A 74 TMACS/W CMOS-RRAM neurosynaptic core with dynamically reconfigurable dataflow and in-situ transposable weights for probabilistic graphical models". In: *2020 IEEE International Solid-State Circuits Conference-(ISSCC)*. IEEE. 2020, pp. 498–500.
- [33] Xue, C.-X., Huang, T.-Y., Liu, J.-S., et al. "A 22nm 2Mb ReRAM compute-in-memory macro with 121-28TOPS/W for multibit MAC computing for tiny AI edge devices". In: *2020 IEEE International Solid-State Circuits Conference-(ISSCC)*. IEEE. 2020, pp. 244–246.
- [34] Yoon, J.-H., Chang, M., Khwa, W.-S., et al. "A 40nm 100Kb 118.44 TOPS/W Ternary-weight Compute-in-Memory RRAM Macro with Voltage-sensing Read and Write Verification for reli-

- able multi-bit RRAM operation". In: *2021 IEEE Custom Integrated Circuits Conference (CICC)*. IEEE. 2021, pp. 1–2.
- [35] He, W., Yin, S., Kim, Y., et al. "2-bit-per-cell RRAM-based in-memory computing for area-/energy-efficient deep learning". In: *IEEE Solid-State Circuits Letters* 3 (2020), pp. 194–197.
- [36] Yoon, J.-H., Chang, M., Khwa, W.-S., et al. "29.1 A 40nm 64Kb 56.67TOPS/W Read-Disturb-Tolerant Compute-in-Memory/Digital RRAM Macro with Active-Feedback-Based Read and In-Situ Write Verification". In: *2021 IEEE International Solid-State Circuits Conference (ISSCC)*. Vol. 64. 2021, pp. 404–406. doi: 10.1109/ISSCC42613.2021.9365926.
- [37] Xue, C.-X., Hung, J.-M., Kao, H.-Y., et al. "A 22nm 4Mb 8b-precision ReRAM computing-in-memory macro with 11.91 to 195.7 TOPS/W for tiny AI edge devices". In: *2021 IEEE International Solid-State Circuits Conference (ISSCC)*. Vol. 64. IEEE. 2021, pp. 245–247.
- [38] Stathopoulos, S., Khiat, A., Trapatseli, M., et al. "Multibit memory operation of metal-oxide bi-layer memristors". In: *Scientific reports* 7.1 (2017), pp. 1–7.
- [39] Maheshwari, S., Stathopoulos, S., Wang, J., et al. "Design Flow for Hybrid CMOS/Memristor Systems—Part I: Modeling and Verification Steps". In: *IEEE Transactions on Circuits and Systems I: Regular Papers* 68.12 (2021), pp. 4862–4875. doi: 10.1109/TCSI.2021.3122343.
- [40] *ArC ONE Memristor Characterisation Platform*. URL: https://files.arc-instruments.co.uk/documents/ArC_One.pdf (visited on 11/29/2024).
- [41] Giotis, C., Serb, A., Stathopoulos, S., et al. "Bidirectional Volatile Signatures of Metal–Oxide Memristors—Part I: Characterization". In: *IEEE Transactions on Electron Devices* 67.11 (2020), pp. 5158–5165.
- [42] Zhao, M., Wu, H., Gao, B., et al. "Investigation of statistical retention of filamentary analog RRAM for neuromorphic computing". In: *2017 IEEE International Electron Devices Meeting (IEDM)*. IEEE. 2017, pp. 39–4.
- [43] Perez, E., Zambelli, C., Mahadevaiah, M. K., et al. "Toward reliable multi-level operation in RRAM arrays: Improving post-algorithm stability and assessing endurance/data retention". In: *IEEE Journal of the Electron Devices Society* 7 (2019), pp. 740–747.
- [44] Kvatinsky, S., Friedman, E. G., Kolodny, A., et al. "TEAM: Threshold Adaptive Memristor Model". In: *IEEE Transactions on Circuits and Systems I: Regular Papers* 60.1 (Jan. 2013). Conference Name: IEEE Transactions on Circuits and Systems I: Regular Papers, pp. 211–221. ISSN: 1558-0806. doi: 10.1109/TCSI.2012.2215714.
- [45] Kvatinsky, S., Ramadan, M., Friedman, E. G., et al. "VTEAM: A General Model for Voltage-Controlled Memristors". en. In: *IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS* 62.8 (2015), p. 5.
- [46] Messaris, I., Serb, A., Stathopoulos, S., et al. "A data-driven verilog-a rram model". In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 37.12 (2018), pp. 3151–3162.
- [47] Chang, T., Jo, S.-H., Kim, K.-H., et al. "Synaptic behaviors and modeling of a metal oxide memristive device". en. In: *Applied Physics A* 102.4 (Mar. 2011), pp. 857–863. ISSN: 1432-0630. doi: 10.1007/s00339-011-6296-1.
- [48] Berdan, R., Vasilaki, E., Khiat, A., et al. "Emulating short-term synaptic dynamics with memristive devices". en. In: *Scientific Reports* 6.1 (Jan. 2016). Number: 1 Publisher: Nature Publishing Group, p. 18639. ISSN: 2045-2322. doi: 10.1038/srep18639.
- [49] Chen, L., Li, C., Huang, T., et al. "A synapse memristor model with forgetting effect". en. In: *Physics Letters A* 377.45 (Dec. 2013), pp. 3260–3265. ISSN: 0375-9601. doi: 10.1016/j.physleta.2013.10.024.

- [50] Wang, Z., Joshi, S., Savel'ev, S. E., et al. "Memristors with diffusive dynamics as synaptic emulators for neuromorphic computing". en. In: *Nature Materials* 16.1 (Jan. 2017), pp. 101–108. ISSN: 1476-1122, 1476-4660. doi: 10.1038/nmat4756.
- [51] Ascoli, A., Corinto, F., Senger, V., et al. "Memristor model comparison". In: *IEEE Circuits and Systems Magazine* 13.2 (2013), pp. 89–105.
- [52] Ascoli, A., Tetzlaff, R., Biolek, Z., et al. "The art of finding accurate memristor model solutions". In: *IEEE Journal on Emerging and Selected Topics in Circuits and Systems* 5.2 (2015), pp. 133–142.
- [53] Kang, S. M., Choi, D., Eshraghian, J. K., et al. "How to build a memristive integrate-and-fire model for spiking neuronal signal generation". In: *IEEE Transactions on Circuits and Systems I: Regular Papers* 68.12 (2021), pp. 4837–4850.
- [54] Demirkol, A. S., Ascoli, A., Messaris, I., et al. "A Compact and Continuous Reformulation of the Strachan TaO Memristor Model With Improved Numerical Stability". In: *IEEE Transactions on Circuits and Systems I: Regular Papers* (2021).
- [55] Ntinis, V., Ascoli, A., Messaris, I., et al. "Toward Simplified Physics-Based Memristor Modeling of Valence Change Mechanism Devices". In: *IEEE Transactions on Circuits and Systems II: Express Briefs* 69.5 (2022), pp. 2473–2477.
- [56] Ingino, J. and Wooley, B. "A continuously-calibrated 10 M sample/s 12 b 3.3 V ADC". In: *1998 IEEE International Solid-State Circuits Conference. Digest of Technical Papers, ISSCC. First Edition (Cat. No.98CH36156)*. 1998, pp. 144–145. doi: 10.1109/ISSCC.1998.672409.
- [57] Dyer, K., Fu, D., Lewis, S., et al. "Analog background calibration of a 10 b 40 Msample/s parallel pipelined ADC". In: *1998 IEEE International Solid-State Circuits Conference. Digest of Technical Papers, ISSCC. First Edition (Cat. No.98CH36156)*. 1998, pp. 142–143. doi: 10.1109/ISSCC.1998.672408.
- [58] Panigada, A. and Galton, I. "A 130 mW 100 MS/s Pipelined ADC With 69 dB SNDR Enabled by Digital Harmonic Distortion Correction". In: *IEEE Journal of Solid-State Circuits* 44.12 (2009), pp. 3314–3328. doi: 10.1109/JSSC.2009.2032637.
- [59] Ding, M., Harpe, P., Liu, Y.-H., et al. "A 46 μ W 13 b 6.4 MS/s SAR ADC With Background Mismatch and Offset Calibration". In: *IEEE Journal of Solid-State Circuits* 52.2 (2017), pp. 423–432. doi: 10.1109/JSSC.2016.2609849.
- [60] Jie, L., Tang, X., Liu, J., et al. "An Overview of Noise-Shaping SAR ADC: From Fundamentals to the Frontier". In: *IEEE Open Journal of the Solid-State Circuits Society* 1 (2021), pp. 149–161. doi: 10.1109/OJSSCS.2021.3119910.
- [61] Si, Z., Wang, C., Malik, A., et al. "Memristor-assisted Background Calibration for Analog-to-Digital Converter". In: *2022 20th IEEE Interregional NEWCAS Conference (NEWCAS)*. 2022, pp. 470–474. doi: 10.1109/NEWCAS52662.2022.9842108.
- [62] Chen, W.-H., Li, K.-X., Lin, W.-Y., et al. "A 65nm 1Mb nonvolatile computing-in-memory ReRAM macro with sub-16ns multiply-and-accumulate for binary DNN AI edge processors". In: *2018 IEEE International Solid-State Circuits Conference-(ISSCC)*. IEEE. 2018, pp. 494–496.
- [63] Wu, T. F., Le, B. Q., Radway, R., et al. "14.3 A 43pJ/Cycle non-volatile microcontroller with 4.7 μ s shutdown/wake-up integrating 2.3-bit/cell resistive RAM and resilience techniques". In: *2019 IEEE International Solid-State Circuits Conference-(ISSCC)*. IEEE. 2019, pp. 226–228.
- [64] Liu, Q., Gao, B., Yao, P., et al. "33.2 A fully integrated analog ReRAM based 78.4 TOPS/W compute-in-memory chip with fully parallel MAC computing". In: *2020 IEEE International Solid-State Circuits Conference-(ISSCC)*. IEEE. 2020, pp. 500–502.
- [65] Yoon, J.-H., Chang, M., Khwa, W.-S., et al. "29.1 A 40nm 64Kb 56.67 TOPS/W read-disturb-tolerant compute-in-memory/digital RRAM macro with active-feedback-based read and in-

- situ write verification". In: *2021 IEEE International Solid-State Circuits Conference (ISSCC)*. Vol. 64. IEEE. 2021, pp. 404–406.
- [66] Xue, C.-X., Hung, J.-M., Kao, H.-Y., et al. "16.1 a 22nm 4mb 8b-precision reram computing-in-memory macro with 11.91 to 195.7 tops/w for tiny ai edge devices". In: *2021 IEEE International Solid-State Circuits Conference (ISSCC)*. Vol. 64. IEEE. 2021, pp. 245–247.
- [67] Zidan, M. A., Strachan, J. P., and Lu, W. D. "The future of electronics based on memristive systems". In: *Nature electronics* 1.1 (2018), pp. 22–29.
- [68] Liu, W., Huang, P., and Chiu, Y. "A 12-bit, 45-MS/s, 3-mW redundant successive-approximation-register analog-to-digital converter with digital calibration". In: *IEEE Journal of Solid-State Circuits* 46.11 (2011), pp. 2661–2672.
- [69] Zhou, Y., Xu, B., and Chiu, Y. "A 12 bit 160 MS/s two-step SAR ADC with background bit-weight calibration using a time-domain proximity detector". In: *IEEE Journal of Solid-State Circuits* 50.4 (2015), pp. 920–931.
- [70] Shen, J., Shikata, A., Fernando, L. D., et al. "A 16-bit 16-MS/s SAR ADC with on-chip calibration in 55-nm CMOS". In: *IEEE Journal of Solid-State Circuits* 53.4 (2018), pp. 1149–1160.
- [71] Cai, F., Correll, J. M., Lee, S. H., et al. "A fully integrated reprogrammable memristor-CMOS system for efficient multiply-accumulate operations". In: *Nature Electronics* 2.7 (2019), pp. 290–299. ISSN: 2520-1131. doi: 10.1038/s41928-019-0270-x.
- [72] Zhu, Z., Zhou, X., Du, Y., et al. "A 14-bit 4-MS/s VCO-Based SAR ADC With Deep Metastability Facilitated Mismatch Calibration". In: *IEEE Journal of Solid-State Circuits* 55.6 (2020), pp. 1565–1576. doi: 10.1109/JSSC.2019.2950188.
- [73] Abo, A. and Gray, P. "A 1.5-V, 10-bit, 14.3-MS/s CMOS pipeline analog-to-digital converter". In: *IEEE Journal of Solid-State Circuits* 34.5 (1999), pp. 599–606. doi: 10.1109/4.760369.
- [74] Harpe, P. J. A., Zhou, C., Bi, Y., et al. "A 26 μ W 8 bit 10 MS/s Asynchronous SAR ADC for Low Energy Radios". In: *IEEE Journal of Solid-State Circuits* 46.7 (2011), pp. 1585–1595. doi: 10.1109/JSSC.2011.2143870.
- [75] Liu, C.-C., Kuo, C.-H., and Lin, Y.-Z. "A 10 bit 320 MS/s Low-Cost SAR ADC for IEEE 802.11ac Applications in 20 nm CMOS". In: *IEEE Journal of Solid-State Circuits* 50.11 (2015), pp. 2645–2654. doi: 10.1109/JSSC.2015.2466475.
- [76] Nakagome, Y., Tanaka, H., Takeuchi, K., et al. "An experimental 1.5-V 64-Mb DRAM". In: *IEEE Journal of Solid-State Circuits* 26.4 (1991), pp. 465–472. doi: 10.1109/4.75040.
- [77] *Coherent Sampling vs. Window Sampling (Analog Devices)*. 2002. URL: <https://www.analog.com/en/technical-articles/coherent-sampling-vs-window-sampling.html>.
- [78] *Histogram Testing Determines DNL and INL Errors (Analog Devices)*. 2003. URL: <https://www.analog.com/en/resources/technical-articles/histogram-testing-determines-dnl-and-inl-errors.html>.
- [79] *Dynamic Testing of High-Speed ADCs, Part 2 (Analog Devices)*. 2002. URL: <https://www.analog.com/en/resources/technical-articles/dynamic-testing-of-highspeed-adcs-part-2.html>.
- [80] Stathopoulos, S., Khiat, A., Trapatseli, M., et al. "Multibit memory operation of metal-oxide bi-layer memristors". In: *Scientific Reports* 7.1 (2017). ISSN: 2045-2322. doi: 10.1038/s41598-017-17785-1.
- [81] Fraz, H. *FPGA DESIGN FOR CAPTURING DATA FROM TI ADCs*. URL: <https://e2e.ti.com/> (visited on 11/29/2024).
- [82] *Zynq™ Evaluation and Development Hardware User's Guide*. URL: https://digilent.com/reference/_media/zedboard:zedboard_ug.pdf?srltid=AfmBOorwbx2zsmgYcQG9pWFZ6-Ut1VZ6PC8ZTDtTNH-iot3P3YDQ9kii (visited on 11/29/2024).

- [83] Mifsud, A., Shen, J., Feng, P., et al. "A CMOS-based Characterisation Platform for Emerging RRAM Technologies". In: *2022 IEEE International Symposium on Circuits and Systems (ISCAS)*. 2022.
- [84] Si, Z., Wang, C., Jiang, X., et al. "Memristor-Assisted CDAC Background Calibration Scheme for SAR ADCs". In: *2025 IEEE International Symposium on Circuits and Systems (ISCAS)*. 2025, pp. 1–5. doi: 10.1109/ISCAS56072.2025.11043236.
- [85] Ding, M., Li, H., and Harpe, P. "Method of offset calibration in a successive approximation register analog-to-digital converter and a successive approximation register analog-to-digital converter". US10230386B2. Mar. 2019.
- [86] Ding, M., Harpe, P., Liu, Y.-H., et al. "26.2 A 5.5fJ/conv-step 6.4MS/S 13b SAR ADC utilizing a redundancy-facilitated background error-detection-and-correction scheme". In: *2015 IEEE International Solid-State Circuits Conference - (ISSCC) Digest of Technical Papers*. 2015, pp. 1–3. doi: 10.1109/ISSCC.2015.7063125.
- [87] Li, Z., Berti, L., Zhao, J., et al. "An 80 MS/s 70.8 dB-SNDR Radiation-Tolerant Semi-Time-Interleaved Pipelined-SAR ADC for Space Applications". In: *IEEE Transactions on Circuits and Systems I: Regular Papers* 72.8 (2025), pp. 3897–3910. doi: 10.1109/TCSI.2024.3516475.
- [88] Li, Z., Berti, L., Lin, Q., et al. "An 80MS/s 70.79dB-SNDR 60.7fJ/Conv-Step Radiation-Tolerant Semi-Time-interleaved Pipelined-SAR ADC". In: *2024 IEEE Custom Integrated Circuits Conference (CICC)*. 2024, pp. 1–2. doi: 10.1109/CICC60959.2024.10529093.
- [89] Harpe, P. "A Compact 10-b SAR ADC With Unit-Length Capacitors and a Passive FIR Filter". In: *IEEE Journal of Solid-State Circuits* 54.3 (2019), pp. 636–645. doi: 10.1109/JSSC.2018.2878830.
- [90] Bodnar, R., Kennedy, H., Hurrell, C. P., et al. "9.8 A 9.3nV/rtHz 20b 40MS/s 94.2dB DR Signal-Chain Friendly Precision SAR Converter". In: *2024 IEEE International Solid-State Circuits Conference (ISSCC)*. Vol. 67. 2024, pp. 182–184. doi: 10.1109/ISSCC49657.2024.10454329.
- [91] Li, H., Shen, Y., Cantatore, E., et al. "Small-Area SAR ADCs With a Compact Unit-Length DAC Layout". In: *IEEE Transactions on Circuits and Systems II: Express Briefs* 69.10 (2022), pp. 4038–4042. doi: 10.1109/TCSII.2022.3186064.
- [92] Liu, C.-C., Chang, S.-J., Huang, G.-Y., et al. "A 10-bit 50-MS/s SAR ADC With a Monotonic Capacitor Switching Procedure". In: *IEEE Journal of Solid-State Circuits* 45.4 (2010), pp. 731–740. doi: 10.1109/JSSC.2010.2042254.
- [93] Tsai, J.-H., Wang, H.-H., Yen, Y.-C., et al. "A 0.003 mm² 10 b 240 MS/s 0.7 mW SAR ADC in 28 nm CMOS With Digital Error Correction and Correlated-Reversed Switching". In: *IEEE Journal of Solid-State Circuits* 50.6 (2015), pp. 1382–1398. doi: 10.1109/JSSC.2015.2413850.
- [94] Huang, G., Wang, C., Si, Z., et al. "An Improved Data-Driven Memristor Model Accounting for Sequences Stimulus Features". In: *2023 IEEE International Symposium on Circuits and Systems (ISCAS)*. 2023, pp. 1–5. doi: 10.1109/ISCAS46773.2023.10182200.

Appendices

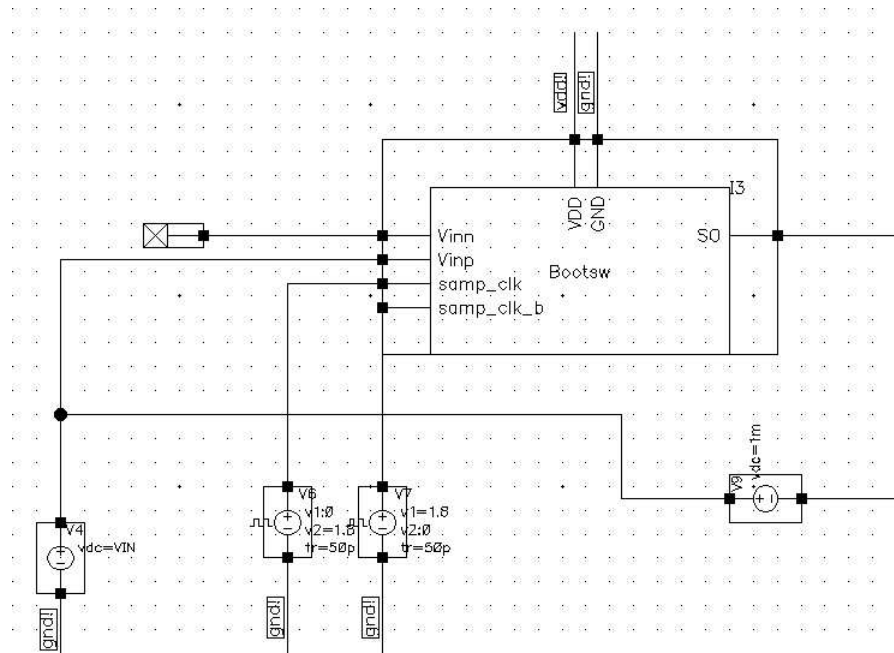


Figure 5.1: Testbench for measuring the on-resistance of the bootstrapped switches.

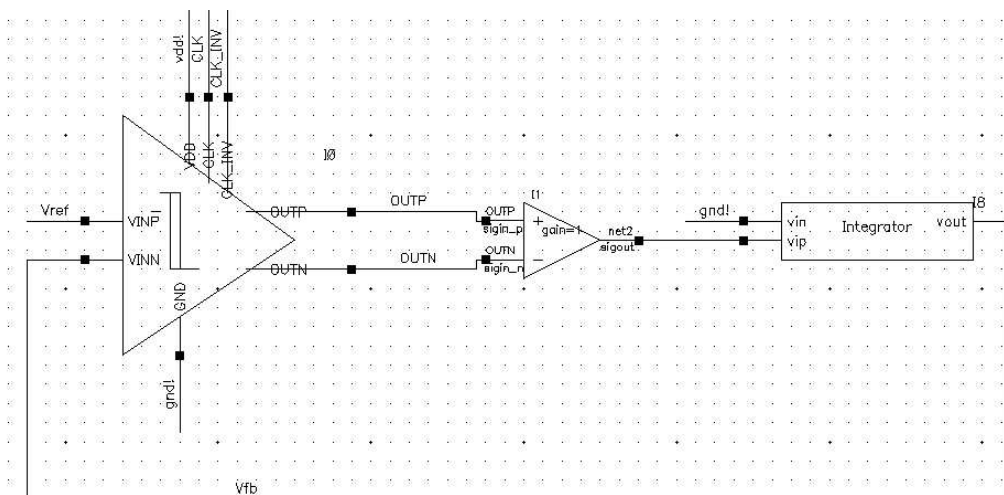
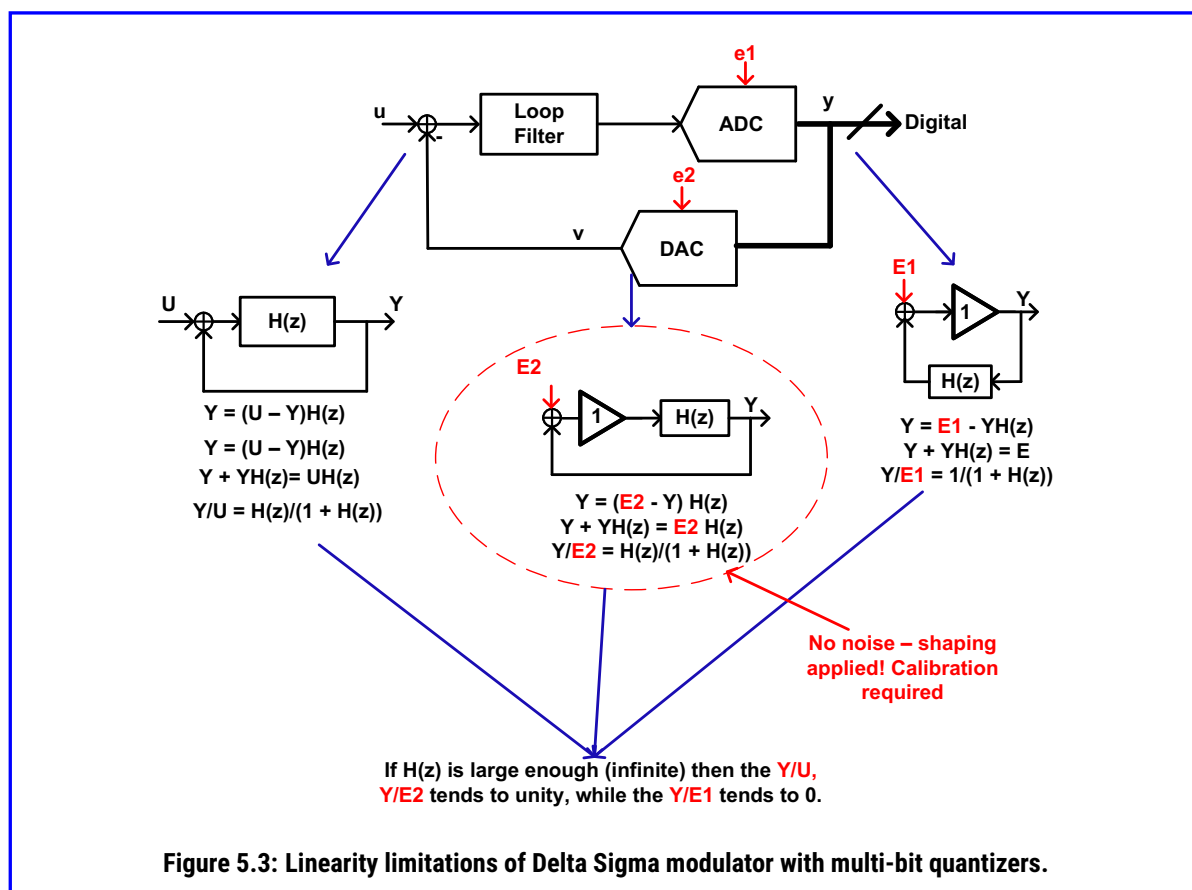


Figure 5.2: Testbench for measuring the input offset of the comparator.



Listing 5.1: MATLAB code for ADC static linearity measurement.

```

1  % Time: 04/2023
2  % Using this script must include necessary citation, the authority of the
3  % code belongs to the authors Dr. Shiwei Wang and Zhaoguang Si.
4  function [histobj] = inldnl_adc_modi(filenameA,filenameD,binsize,nsample)
5  %filenameA: name of the CSV file containing analogue input data;
6  %filenameD: name of the CSV file containing ADC output data;
7  %binsize: size of the bin (number of points in each conversion step);
8  %nsample: number of samples/points from the simulation. Ideally making
9  %nsample/binsize an integer which represents the number of bins (codes)
10 %covered.
11
12 %Omitted the initial 100 codes
13 ain=csvread(filenameA, 40*binsize+1, 1, [40*binsize+1 1 nsample 1]);
14 dout=csvread(filenameD,40*binsize+1, 1, [40*binsize+1 1 nsample 1]);
15
16 ncode=nsample/binsize - 40; %Omitted the initial 100 codes
17
18 hisobj=histogram(dout,ncode);
19 dnl=hisobj.Values/binsize-1;
20 inl=cumsum(dnl);
21 lsb=(max(ain)-min(ain))/ncode;
22 code=linspace(min(ain)/lsb,max(ain)/lsb,ncode);
23
24 plot(code,dnl);
25 grid on;
26 title('DNL')
27 xlabel('Digital Code')
28 ylabel('DNL/LSB')
29
30 figure;
31 plot(code,inl);
32 grid on;
33 title('INL')
34 xlabel('Digital Code')
35 ylabel('INL/LSB')
36 end

```

Listing 5.2: VerilogA code for ramp generator.

```

1 // VerilogA for Memristor_assisted_ADC_Design_V2, Ramp_Signal_Generator, verilogA
2
3 include "constants.vams"
4 include "disciplines.vams"
5
6 module Ramp_Signal_Generator(vramp, clk);
7 // vramp, (output) input voltage for comparator, non-inverting
8 // clk, clock signal for the comparator
9 input clk ;
10 output vramp ;
11 electrical vramp, clk ;
12
13 parameter vth=1 ; // clk threshold voltage
14 parameter resolution=100u ; // resolution of the ramp
15 parameter maxin = 0.8 ; // maximum input level
16 parameter minin = 0.5 ;
17 parameter td = 100p ; // delay time
18 parameter tr = 100p ; // rise time
19 parameter initdirection = 1;
20
21 real offset ;
22 integer direction ;
23
24 analog
25 begin
26   @(initial_step("static","tran")) begin // On initialization,
27     if (initdirection > 0) begin
28       offset = minin;
29       direction = 1 ;
30     end
31     else begin
32       offset = maxin;
33       direction = -1 ;
34     end
35   end
36
37   @( cross( V(clk)>vth, -1 )) begin
38     offset = offset + (direction * resolution) ;
39   end
40
41   if(offset > maxin-resolution/2) begin
42     direction = -1 ;
43   end
44   else if(offset < minin+resolution/2) begin
45     direction = 1;
46   end
47
48   V(vramp) <+ transition( offset, td, tr, tr ) ;
49
50 end
51
52 endmodule

```

Listing 5.3: MATLAB code for ADC dynamic linearity test [79].

```

1 %The original codes are from Analog Devices Application Note Dynamic Testing of
2 %High-Speed ADCs, Part 2, but with minor modifications to meet the use of this
3 %design. Please add proper citation when using this code.
4
5 numpt = input('Data Record Size (Number of Points)?');
6 fclk = input('Sampling Frequency (MHz)?');
7 fres = fclk/numpt;
8 numbit = 13; %can be changed based on the DUT's resolution
9
10 data1 = importdata('bootsp_sw_test_data.csv');
11
12 timedomaindata = data1(:, 2); %in csv file the second column
13
14 if (max(timedomaindata)==2^numbit-1)||(min(timedomaindata)==0)
15     disp('Warning, ADC may be Clipping!!!')
16 end
17
18 figure(1);
19 plot([1:numpt],timedomaindata);
20 title('Time Domain');
21 xlabel('Samples');
22 ylabel('Digital Output Code');
23
24
25
26 fin_dB=20*log10((max(timedomaindata)-min(timedomaindata))/(2^numbit));
27 offset = mean (timedomaindata);
28 Dout = timedomaindata-offset; %- (2^numbit - 1)/2;
29
30 Doutw = Dout.*hanning(numpt);
31
32 Dout_spect = fft(Doutw);
33 Dout_dB=20*log10(abs(Dout_spect));
34
35 figure(2);
36 maxdB = max(Dout_dB(1:numpt/2));
37 plot([0:numpt/2-1].*fclk/numpt, Dout_dB(1:numpt/2)-maxdB);
38 grid on;
39 title('FFT');
40 xlabel('Analog Input Frequency (MHz)');
41 ylabel('Amplitude (dB)');
42
43 spandc=0; %Approximate search span for DC on right side
44 %Calculate SNR, SINAD, THD and SFDR values
45 %Find the signal bin number, DC = bin 1
46 fin=find(Dout_dB(1:numpt/2)==maxdB);
47 %Span of the input frequency on each side
48 span=max(round(numpt/200),10);
49 %Approximate search span for harmonics on each side
50 spanh=8;
51 %Determine power spectrum
52 spectP=(abs(Dout_spect)).*(abs(Dout_spect));
53 %Find DC offset power
54 Pdc=abs(sum(spectP(1:span)));
55 %Extract overall signal power
56 idx1 = fin - span;
57 idx2 = fin + span;
58

```

```

59  if(idx1<=0)
60      idx1 = 1;
61  end
62
63  Ps=abs(sum(spectP(idx1:idx2)));
64  %Vector/matrix to store both frequency and power of signal and harmonics
65  Fh=[];
66  %The 1st element in the vector/matrix represents the signal, the next element represents
67  %the 2nd harmonic, etc.
68  Ph=[];
69  Nh=[];
70  %Ah represents signal and harmonic amplitude
71  Ah=[];
72  %Find harmonic frequencies and power components in the FFT spectrum
73  %For this procedure to work, ensure the folded back high order harmonics do not overlap
74  %with DC or signal or lower order harmonics , so it should be modified according to the actual
  ↪ condition
75
76  for har_num=1:5
77      tone=rem((har_num*(fin-1)+1)/numpt,1); %Input tones greater than fSAMPLE are aliased back
  ↪ into the spectrum
78      if tone>0.5
79          tone=1-tone; %Input tones greater than 0.5*Fsample (after aliasing) are reflected
80      end
81      Fh=[Fh tone];
82      %Check Nh to see the bin of the harmonics
83      Nh=[Nh round(tone*numpt)];
84      %For this procedure to work, ensure the folded back high order harmonics do not overlap
85      %with DC or signal or lower order harmonics
86      idx3 = round(tone*numpt)-spanh;
87      if(idx3<=0)
88          idx3 = 1;
89      end
90      har_peak=max(spectP(idx3:round(tone*numpt)+spanh));
91      har_bin=find(spectP(idx3:round(tone*numpt)+spanh)==har_peak);
92      har_bin=har_bin+round(tone*numpt)-spanh-1;
93      idx4 = har_bin-spanh;
94      if(idx4<=0)
95          idx4 = 1;
96      end
97      Ph=[Ph sum(spectP(idx4:har_bin+spanh))];
98      Ah=[Ah Dout_dB(abs(har_bin))];
99  end
100
101  %Determine the total distortion power, it should be modified according to the actual condition.
102  Pd=abs(sum(Ph(2:5)));
103  %Determine the noise power
104  Pn=abs(sum(spectP(1:numpt/2))-Pdc-Ps-Pd);
105  %Determine the next largest component
106  spur_max=max(max(spectP(spandc+1:abs(fin-span-1))),max(spectP(fin+span+1:numpt/2)));
107  SFDR = abs(10*log10(max(spectP(1:numpt/2))/spur_max)-fin_dB);
108  %SFDR = abs(10*log10(max(spectP(1:numpt/2))/spur_max));
109  THD = 10*log10(Pd/Ps)+fin_dB;
110  %THD = 10*log10(Pd/Ps);
111  SNR = abs(10*log10(Ps/Pn)-fin_dB);
112  %SNR = 10*log10(Ps/Pn);
113  SNDR = abs(10*log10(Ps/(Pn+Pd))-fin_dB);
114  %SNDR = 10*log10(Ps/(Pn+Pd));
115  ENOB = (SNDR-1.76)/6.02;
116
117  ylabel('Full-Scale Normalized Magnitude[dB]');
118  xlabel('Frequency [MHz]');

```

```
119 grid on;
120 box on;
121 hold on;
122 s1=sprintf('SFDR = %4.1fdB\n',SFDR);
123 s2=sprintf('THD = %4.1fdB\n',THD);
124 s3=sprintf('SNR = %4.1fdB\n',SNR);
125 s4=sprintf('SNDR = %4.1fdB\n',SNDR);
126 s5=sprintf('ENOB = %4.2fbit\n',ENOB);
127 text(0.04,-10,s1);
128 text(0.04,-20,s2);
129 text(0.04,-30,s3);
130 text(0.04,-40,s4);
131 text(0.04,-50,s5);
132
133
134
135
136
137
```

Listing 5.4: ADC interface module Verilog code

```

1 module adc_interface (
2     // System interface
3     input wire clk,           // System clock
4     input wire rst_n,        // Active low reset
5
6     // ADC interface (12 individual pins)
7     input wire [11:0] adc_data_pins, // Individual ADC data pins
8     input wire adc_valid, // ADC data valid signal
9
10    // AXI-Lite interface for Zynq processor
11    input wire s_axi_aclk,
12    input wire s_axi_aresetn,
13    input wire [31:0] s_axi_awaddr,
14    input wire s_axi_awvalid,
15    output wire s_axi_awready,
16    input wire [31:0] s_axi_wdata,
17    input wire [3:0] s_axi_wstrb,
18    input wire s_axi_wvalid,
19    output wire s_axi_wready,
20    output wire [1:0] s_axi_bresp,
21    output wire s_axi_bvalid,
22    input wire s_axi_bready,
23    input wire [31:0] s_axi_araddr,
24    input wire s_axi_arvalid,
25    output wire s_axi_arready,
26    output wire [31:0] s_axi_rdata,
27    output wire [1:0] s_axi_rresp,
28    output wire s_axi_rvalid,
29    input wire s_axi_rready
30 );
31
32 // Parameters
33 localparam FIFO_DEPTH = 1024;
34 localparam ADC_WIDTH = 12;
35
36 // Internal registers
37 reg [ADC_WIDTH-1:0] adc_data_reg;
38 reg [9:0] write_pointer;
39 reg [9:0] read_pointer;
40 reg fifo_full;
41 reg fifo_empty;
42
43 // Memory for storing ADC samples
44 reg [ADC_WIDTH-1:0] sample_memory [0:FIFO_DEPTH-1];
45
46 // Control and status registers
47 reg [31:0] control_reg; // Bit 0: Enable sampling
48 // Bit 1: Clear FIFO
49 // Bit 2: Interrupt enable
50 reg [31:0] status_reg; // Bit 0: FIFO full
51 // Bit 1: FIFO empty
52 // Bit 2: Data ready
53
54 // Sample counter and debug registers
55 reg [31:0] sample_count;
56 reg [3:0] debug_state;
57
58 // Test signal generator

```

```

59     reg [15:0] valid_counter;
60     reg test_valid;
61     reg [11:0] test_data_counter;
62
63     // Clock domain crossing signals
64     reg [2:0] adc_valid_sync;
65     wire adc_valid_pedge;
66     reg adc_valid_caught;
67
68     // AXI interface registers
69     reg [31:0] axi_rdata;
70     reg axi_rvalid;
71     reg axi_arready;
72     reg axi_awready;
73     reg axi_wready;
74     reg axi_bvalid;
75
76     // Initialize registers
77     initial begin
78         fifo_empty = 1'b1;
79         fifo_full = 1'b0;
80         write_pointer = 10'h0;
81         read_pointer = 10'h0;
82         sample_count = 32'h0;
83         control_reg = 32'h0;
84         status_reg = 32'h0;
85         debug_state = 4'h0;
86         valid_counter = 16'h0;
87         test_valid = 1'b0;
88         test_data_counter = 12'h0;
89         adc_valid_caught = 1'b0;
90         adc_valid_sync = 3'b000;
91     end
92
93     // Generate test valid signal (about 100kHz assuming 100MHz clock)
94     always @(posedge clk or negedge rst_n) begin
95         if (!rst_n) begin
96             valid_counter <= 16'h0;
97             test_valid <= 1'b0;
98         end else if (control_reg[0]) begin // Only when ADC is enabled
99             if (valid_counter == 16'd999) begin // Divide by 1000
100                 valid_counter <= 16'h0;
101                 test_valid <= ~test_valid;
102             end else begin
103                 valid_counter <= valid_counter + 1'b1;
104             end
105         end
106     end
107
108     // Clock domain crossing
109     always @(posedge clk or negedge rst_n) begin
110         if (!rst_n) begin
111             adc_valid_sync <= 3'b000;
112             adc_valid_caught <= 1'b0;
113         end else begin
114             adc_valid_sync <= {adc_valid_sync[1:0], test_valid}; // Use test_valid instead of
115                 ↪ adc_valid
116             adc_valid_caught <= adc_valid_sync[1];
117         end
118     end
119
120     assign adc_valid_pedge = adc_valid_sync[1] & ~adc_valid_sync[2];

```

```

120
121 // Test data counter
122 always @(posedge clk or negedge rst_n) begin
123     if (!rst_n) begin
124         test_data_counter <= 12'h0;
125     end else if (adc_valid_pedge && control_reg[0]) begin
126         test_data_counter <= test_data_counter + 1'b1;
127     end
128 end
129
130 // ADC data capture and FIFO management
131 always @(posedge clk or negedge rst_n) begin
132     if (!rst_n) begin
133         adc_data_reg <= {ADC_WIDTH{1'b0}};
134         write_pointer <= 10'h000;
135         read_pointer <= 10'h000;
136         fifo_full <= 1'b0;
137         fifo_empty <= 1'b1;
138         sample_count <= 32'h0;
139     end else begin
140         if (control_reg[1]) begin // FIFO clear
141             write_pointer <= 10'h000;
142             read_pointer <= 10'h000;
143             fifo_full <= 1'b0;
144             fifo_empty <= 1'b1;
145             sample_count <= 32'h0;
146         end else begin
147             // Write operation
148             if (adc_valid_pedge && control_reg[0] && !fifo_full) begin
149                 adc_data_reg <= test_data_counter; // Use test data
150                 sample_memory[write_pointer] <= test_data_counter;
151                 write_pointer <= write_pointer + 1'b1;
152                 sample_count <= sample_count + 1'b1;
153                 fifo_empty <= 1'b0;
154
155                 if ((write_pointer + 1'b1) == read_pointer)
156                     fifo_full <= 1'b1;
157             end
158
159             // Read operation
160             if (s_axi_arvalid && !axi_rvalid &&
161                 s_axi_araddr[7:0] == 8'h0C && !fifo_empty) begin
162                 read_pointer <= read_pointer + 1'b1;
163                 fifo_full <= 1'b0;
164
165                 if ((read_pointer + 1'b1) == write_pointer)
166                     fifo_empty <= 1'b1;
167             end
168         end
169     end
170 end
171
172 // Debug state update
173 always @(posedge clk or negedge rst_n) begin
174     if (!rst_n)
175         debug_state <= 4'h0;
176     else
177         debug_state <= {
178             test_valid, // Using test_valid instead of adc_valid
179             adc_valid_pedge,
180             control_reg[0],
181             fifo_full

```

```

182         };
183     end
184
185     // Status register update
186     always @(posedge clk or negedge rst_n) begin
187         if (!rst_n) begin
188             status_reg <= 32'h0;
189         end else begin
190             if (adc_valid_pedge && control_reg[0] && !fifo_full) begin
191                 status_reg[2] <= 1'b1; // Set data ready when new data arrives
192             end else if (s_axi_arvalid && s_axi_araddr[7:0] == 8'h0C) begin
193                 status_reg[2] <= 1'b0; // Clear data ready when data is read
194             end
195
196             status_reg[1] <= fifo_empty;
197             status_reg[0] <= fifo_full;
198         end
199     end
200
201     // AXI read interface
202     always @(posedge s_axi_aclk or negedge s_axi_aresetn) begin
203         if (!s_axi_aresetn) begin
204             axi_rdata <= 32'h0;
205             axi_rvalid <= 1'b0;
206             axi_arready <= 1'b0;
207         end else begin
208             if (s_axi_arvalid && !axi_rvalid) begin
209                 case (s_axi_araddr[7:0])
210                     8'h00: axi_rdata <= control_reg;
211                     8'h04: axi_rdata <= status_reg;
212                     8'h08: axi_rdata <= {28'h0, debug_state};
213                     8'h0C: axi_rdata <= {20'h0, sample_memory[read_pointer]};
214                     8'h10: axi_rdata <= write_pointer;
215                     8'h14: axi_rdata <= read_pointer;
216                     default: axi_rdata <= 32'h0;
217                 endcase
218                 axi_rvalid <= 1'b1;
219                 axi_arready <= 1'b1;
220             end else if (axi_rvalid && s_axi_rready) begin
221                 axi_rvalid <= 1'b0;
222                 axi_arready <= 1'b0;
223             end
224         end
225     end
226
227     // AXI write interface
228     always @(posedge s_axi_aclk or negedge s_axi_aresetn) begin
229         if (!s_axi_aresetn) begin
230             control_reg <= 32'h0;
231             axi_awready <= 1'b0;
232             axi_wready <= 1'b0;
233             axi_bvalid <= 1'b0;
234         end else begin
235             if (s_axi_awvalid && s_axi_wvalid && !axi_bvalid) begin
236                 case (s_axi_awaddr[7:0])
237                     8'h00: control_reg <= s_axi_wdata;
238                     default: control_reg <= control_reg;
239                 endcase
240                 axi_awready <= 1'b1;
241                 axi_wready <= 1'b1;
242                 axi_bvalid <= 1'b1;
243             end else if (axi_bvalid && s_axi_bready) begin

```

```
244         axi_bvalid <= 1'b0;
245         axi_awready <= 1'b0;
246         axi_wready <= 1'b0;
247     end
248 end
249 end
250
251 // AXI output assignments
252 assign s_axi_awready = axi_awready;
253 assign s_axi_wready = axi_wready;
254 assign s_axi_bresp = 2'b00;
255 assign s_axi_bvalid = axi_bvalid;
256 assign s_axi_arready = axi_arready;
257 assign s_axi_rdata = axi_rdata;
258 assign s_axi_rresp = 2'b00;
259 assign s_axi_rvalid = axi_rvalid;
260
261 endmodule
```

Listing 5.5: Pin constraints for adc interface module

```

1  # Configuration Settings
2  set_property CONFIG_VOLTAGE 3.3 [current_design]
3  set_property CFGBVS VCCO [current_design]
4
5  # PMOD JA Pins
6  set_property PACKAGE_PIN Y11 [get_ports {adc_data_pins[0]}]
7  set_property PACKAGE_PIN AA11 [get_ports {adc_data_pins[1]}]
8  set_property PACKAGE_PIN Y19 [get_ports {adc_data_pins[2]}]
9  set_property PACKAGE_PIN AA9 [get_ports {adc_data_pins[3]}]
10 set_property IOSTANDARD LVCMOS18 [get_ports {adc_data_pins[0]}]
11 set_property IOSTANDARD LVCMOS18 [get_ports {adc_data_pins[1]}]
12 set_property IOSTANDARD LVCMOS18 [get_ports {adc_data_pins[2]}]
13 set_property IOSTANDARD LVCMOS18 [get_ports {adc_data_pins[3]}]
14
15 # PMOD JA Pins
16 set_property PACKAGE_PIN AB11 [get_ports {adc_data_pins[4]}]
17 set_property PACKAGE_PIN AB10 [get_ports {adc_data_pins[5]}]
18 set_property PACKAGE_PIN AB9 [get_ports {adc_data_pins[6]}]
19 set_property PACKAGE_PIN AA8 [get_ports {adc_data_pins[7]}]
20 set_property IOSTANDARD LVCMOS18 [get_ports {adc_data_pins[4]}]
21 set_property IOSTANDARD LVCMOS18 [get_ports {adc_data_pins[5]}]
22 set_property IOSTANDARD LVCMOS18 [get_ports {adc_data_pins[6]}]
23 set_property IOSTANDARD LVCMOS18 [get_ports {adc_data_pins[7]}]
24
25 # PMOD JB Pins
26 set_property PACKAGE_PIN W12 [get_ports {adc_data_pins[8]}]
27 set_property PACKAGE_PIN W11 [get_ports {adc_data_pins[9]}]
28 set_property PACKAGE_PIN V10 [get_ports {adc_data_pins[10]}]
29 set_property PACKAGE_PIN W8 [get_ports {adc_data_pins[11]}]
30 set_property IOSTANDARD LVCMOS18 [get_ports {adc_data_pins[8]}]
31 set_property IOSTANDARD LVCMOS18 [get_ports {adc_data_pins[9]}]
32 set_property IOSTANDARD LVCMOS18 [get_ports {adc_data_pins[10]}]
33 set_property IOSTANDARD LVCMOS18 [get_ports {adc_data_pins[11]}]
34
35 # ADC Valid Signal
36 set_property PACKAGE_PIN V12 [get_ports adc_valid]
37 set_property IOSTANDARD LVCMOS18 [get_ports adc_valid]
38 # Drive Strength and Slew Rate settings individually
39

```

Listing 5.6: *adc_interface* module software control.

```

#include <stdio.h>
#include "xil_printf.h"
#include "xil_io.h"
#include "xparameters.h"
#include "sleep.h"

#define ADC_BASEADDR      0x43C00000
#define MAX_SAMPLES      4096
#define ADC_BITS          12
#define ADC_MAX_VALUE     ((1 << ADC_BITS) - 1) // 4095 for 12-bit

// Register offsets
#define CONTROL_REG        0x00
#define STATUS_REG         0x04
#define DATA_REG          0x0C

// Control Register Bits
#define ADC_ENABLE         0x01
#define FIFO_CLEAR         0x02

// Status Register Bits
#define DATA_READY        0x04
#define FIFO_EMPTY         0x02
#define FIFO_FULL          0x01

typedef struct {
    uint16_t raw_12bit;    // Store exact 12-bit value
    float voltage;
} ADC_Sample;

ADC_Sample samples[MAX_SAMPLES];

int main()
{
    uint32_t status;
    int sample_count = 0;

```

```

xil_printf("\r\n=== 12-bit ADC Data Collection Started ===\r\n");
xil_printf("ADC Resolution: %d bits (0-%d)\r\n", ADC_BITS, ADC_MAX_VALUE);

// Clear FIFO
Xil_Out32(ADC_BASEADDR + CONTROL_REG, FIFO_CLEAR);
usleep(1000);

// Enable ADC
Xil_Out32(ADC_BASEADDR + CONTROL_REG, ADC_ENABLE);

// Collect samples
while(sample_count < MAX_SAMPLES) {
    status = Xil_In32(ADC_BASEADDR + STATUS_REG);

    if(status & DATA_READY) {
        // Read and mask to ensure 12-bit value
        samples[sample_count].raw_12bit =
            (uint16_t)(Xil_In32(ADC_BASEADDR + DATA_REG) & ADC_MAX_VALUE);

        // Calculate voltage (1.8V reference)
        samples[sample_count].voltage =
            (float)samples[sample_count].raw_12bit * 1.8 / ADC_MAX_VALUE;

        if(sample_count % 512 == 0) {
            xil_printf("Collected %d samples...\r\n", sample_count);
        }

        sample_count++;
    }

    usleep(10); // 10us delay for ~100kS/s sampling
}

// Transmit data in binary format (12-bit)
xil_printf("\r\nSTART_DATA\r\n");
xil_printf("Sample,Raw12Bit,Voltage\r\n");

for(int i = 0; i < MAX_SAMPLES; i++) {
    // Print sample number, raw 12-bit value (in hex and decimal), and voltage

```

```
        xil_printf("%d,0x%03X,%d,%.6f\r\n",
                    i,
                    samples[i].raw_12bit,
                    samples[i].raw_12bit,
                    samples[i].voltage);
        usleep(100);
    }

    xil_printf("END_DATA\r\n");
    xil_printf("Data format: SampleNumber,HexValue,DecimalValue,Voltage\r\n");

    return 0;
}
```